

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA



TESIS DOCTORAL

Algoritmos metaheurísticos para la segmentación de imágenes

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Salvador Miguel Hinojosa Cervantes

Directores

Gonzalo Pajares Martinsanz
Erik Cuevas Jiménez
Diego Oliva Navarro

Madrid
Ed. electrónica 2019

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA



TESIS DOCTORAL

Algoritmos metaheurísticos para la segmentación de imágenes

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Salvador Miguel Hinojosa Cervantes

Directores

Gonzalo Pajares Martinsanz

Erik Cuevas Jiménez

Diego Oliva Navarro

Madrid, 2019

Agradecimientos

Agradezco a mi familia y amigos por su apoyo para lograr esta meta. En especial agradezco los consejos y el buen ejemplo de los Profesores que dirigieron el curso de esta investigación. Además, agradezco al Gobierno de México por financiar los estudios mediante la beca doctoral 298285 del Consejo Nacional de Ciencia y Tecnología (CONACyT).

Salvador Hinojosa

Índice general

Resumen	1
Abstract.....	4
1 Introducción.....	7
1.1 Antecedentes.....	8
1.2 Planteamiento del problema.....	10
1.3 Objetivos.....	11
1.3.1 Objetivos generales.....	11
1.3.2 Objetivos específicos	12
1.4 Metodología.....	12
1.5 Aportaciones de la investigación	13
1.5.1 Artículos publicados en revistas indexadas en el Journal Citation Reports (JCR)	13
1.5.2 Capítulos de libros	16
1.5.3 Congreso.....	17
1.6 Descripción de los capítulos	17
1.7 Resumen	18
2 Marco teórico y estado del arte.....	19
2.1 Optimización.....	19
2.2 Algoritmos metaheurísticos para optimización	21
2.2.1 Algoritmos evolutivos mono-objetivo	22
2.2.2 Algoritmos evolutivos multiobjetivo	22
2.3 Clasificación de algoritmos de evolutivos	23
2.4 Procesamiento de imágenes y visión artificial	27
2.4.1 Procesamiento digital de imágenes	27
2.4.2 Histograma.....	28
2.4.3 Umbralización	32
2.4.4 Varianza entre clases	33
2.4.5 Entropía máxima.....	35
2.4.6 Entropía cruzada mínima	36
2.5 Criterios de comparación	39
2.5.1 MSE.....	39
2.5.2 PSNR.....	39
2.5.3 SSIM.....	39
2.5.4 FSIM.....	40
2.5.5 Prueba de significancia	41
2.6 Resumen	41
3 Umbralización usando entropía cruzada mínima	43
3.1 Descripción del algoritmo EMO	43
3.2 Segmentación de imágenes usando entropía cruzada.	47
3.3 Resultados experimentales	48
3.3.1 Resultados con entropía cruzada.....	49
3.4 Comparaciones.....	54
3.5 Resumen	59
4 Umbralización de imágenes especiales	61
4.1 Resonancias magnéticas.....	61
4.1.1 Descripción del algoritmo CSA.....	62
4.1.2 Umbralización mediante CSA y entropía cruzada	63
4.1.3 Montaje experimental.	65
4.1.4 Resultados experimentales.....	66
4.1.5 Comparaciones	77
4.1.6 Discusión	79
4.2 Imágenes térmicas.....	80
4.2.1 Descripción de algoritmos utilizados.....	82
4.2.2 Montaje experimental	86
4.2.3 Resultados usando la varianza entre clases.....	87

4.2.4	Resultados usando la entropía máxima	89
4.3	Resumen.	92
5	Umbralización basada en contexto	93
5.1	Curva de energía	94
5.2	Segmentación de imágenes sensibles al contexto usando ALO o SCA.	96
5.2.1	Optimizador de la hormiga león (ALO) para UM	97
5.2.2	Algoritmo seno-coseno para UM.....	99
5.3	Resultados experimentales.....	100
5.3.1	Resultados usando la varianza entre clases.....	102
5.3.2	Resultados usando la entropía máxima.....	107
5.3.3	Comparación estadística	112
5.4	Discusión	113
5.5	Resumen	114
6	Segmentación de imágenes con un enfoque multiobjetivo	115
6.1	Descripción de algoritmos multiobjetivo.....	118
6.1.1	NSGA-III.....	118
6.1.2	PESA-II	119
6.1.3	MOPSO	120
6.2	Umbralización sin asistencia (UTH).....	120
6.2.1	Funciones objetivo.....	121
6.2.2	Partícula de longitud variable	122
6.2.3	Operador de cruce modificado.....	123
6.2.4	Proceso de toma de decisiones y segmentación de imagen	124
6.2.5	Resultados experimentales.....	125
6.2.6	Discusión	136
6.3	Mejora de algoritmos multiobjetivo usando caos.	137
6.3.1	CSA multiobjetivo	137
6.3.2	CSA caótico y multiobjetivo.....	139
6.3.3	Resultados experimentales.....	141
6.3.4	Discusión	145
6.4	Resumen	152
7	Conclusiones y trabajo futuro.....	153
7.1	Conclusiones específicas	153
7.1.1	Umbralización con entropía cruzada mínima.	153
7.1.2	Umbralización de imágenes especiales.....	155
7.1.3	Umbralización basado en contexto.	156
7.1.4	Segmentación de imágenes con un enfoque multiobjetivo	157
7.2	Conclusiones generales.....	159
7.3	Trabajo futuro	160
	Bibliografía	161

Índice de abreviaturas

ABC	Colonia artificial de abejas, <i>artificial bee colony</i>
ACOR	Colonia de hormigas para espacios reales, <i>ant colony optimization for real spaces</i>
ALO	Optimizador de la hormiga león, <i>antl-lion optimizer</i>
AM	Algoritmos metaheurísticos
AMT	Umbralización multinivel automático, <i>automatic multilevel thresholding</i>
AP	Algoritmos poblacionales
ATMO	Umbralización automático multiobjetivo, <i>automatic thresholding multiobjective</i>
BA	Algoritmo del murciélago, <i>bat algorithm</i>
CAB	Comportamiento colectivo de animales, <i>collective animal behavior</i>
CSA	Algoritmo de búsqueda del cuervo, <i>crow search algorithm</i>
DE	Evolución diferencial, <i>differential evolution</i>
EA	Algoritmos evolutivos, <i>evolutionary algorithms</i>
ECM	Entropía cruzada mínima
EMO	Optimización basada en magnetismo, <i>electromagnetism-like optimization</i>
FFA	Algoritmo de luciérnagas, <i>firefly algorithm</i>
FLIR	Imágenes térmicas, <i>forward-looking infra red</i>
FPA	Algoritmo de polinización de flores, <i>flower pollination algorithm</i>
FSIM	Índice de similitud de características, <i>feature similarity index measure</i>
GA	Algoritmos genéticos, <i>genetic algorithms</i>
GPU	Unidad de procesamiento gráfico, <i>graphics processing unit</i>
HS	Búsqueda de armonías, <i>harmony search</i>
IGD	Distancia generacional inversa, <i>inverse generational distance</i>
ILS	Búsqueda local iterativa, <i>iterated local search</i>
MOCCSA	CSA caótico y multiobjetivo
MOCSA	CSA multiobjetivo
MOEA	Algoritmos evolutivos multi objetivo, <i>multi-objective evolutionary algorithms</i>
MOEA/D	Algoritmo evolutivo multiobjetivo basado en descomposición, <i>multiobjective evolutionary algorithm based on decomposition</i>
MOPSO	PSO multiobjetivo
MS	Extensión máxima, <i>maximum spread</i>
MSE	Error cuadrático medio, <i>mean squared error</i>
NFL	Teorema <i>no free lunch</i>
NSGA	Algoritmo genético con ordenamiento no dominado, <i>non-dominated sorting genetic algorithm</i>
PESA	Selección basada en Pareto envolvente, <i>Pareto envelope based selection</i>
PSNR	Proporción máxima de señal a ruido, <i>peak signal-to-noise ratio</i>
PSO	Optimización de enjambre de partículas, <i>particle swarm optimization</i>
RM	Resonancia magnética
RRA	Algoritmo de tallos, <i>runner-root algorithm</i>
SA	Forja simulada, <i>simulated annealing</i>
SCA	Algoritmo de seno y coseno, <i>sine-cosine algorithm</i>
SP	Extensión, <i>spread</i>
SPEA	Algoritmo evolutivo basado en la fuerza de Pareto, <i>strenght Pareto evolutionary algorithm</i>
SSA	Algoritmo de la araña social, <i>social spider algorithm</i>
SSIM	Índice de similitud estructural, <i>structural similarity index measure</i>
TS	Búsqueda tabú, <i>tabu search</i>

UI	Umbralización de imágenes
UM	Umbralización multinivel
UTH	Umbralización sin asistencia, <i>unassisted thresholding</i>

Índice de figuras

Figura 2.1. Vecindad del píxel P.	28
Figura 2.2. Imagen y su histograma correspondiente.	29
Figura 2.3. Comparación del efecto de la selección del umbral. a) umbral en 50, b) umbral en 100 y c) umbral en 150.	32
Figura 2.4. Ejemplo de umbralización multinivel con cuatro clases y tres umbrales ($th = [83 \ 127 \ 176]$).	33
Figura 3.1. Imágenes usadas para la comparación visual: (a) Cameraman, (c) Lena, (e) Baboon, (g) Hunter y (i) Butterfly. Los histogramas correspondientes a cada imagen (b), (d), (f), (h), (j).	51
Figura 4.1. Comparación de tiempo Otsu vs. Entropía cruzada usando CSA.	71
Figura 5.1. Representación espacial de sistema de vecindad N^2	95
Figura 5.2. Imagen de ejemplo I, matriz binaria B_i para $l=35$ y $l=80$. Matriz constante.	95
Figura 5.3. Metodología propuesta para la incorporación de información contextual.	96
Figura 5.4. Subconjunto de imágenes de prueba y sus respectivos histogramas y curvas de energía.	101
Figura 5.5. Tiempo promedio de cada algoritmo usando Otsu y 2, 4, 8, 16 y 32 umbrales.	107
Figura 5.6. Tiempo promedio de cada algoritmo usando Kapur y 2, 4, 8, 16 y 32 umbrales. ..	111
Figura 6.1. Representación de partículas en UTH.	123
Figura 6.2. Operador de cruce modificado basado en la cercanía del valor.	124
Figura 6.3. Primer conjunto de imágenes. a) Mmi3, b) Mmi5, c) Rice, d) Tape y, e) Cell.	127
Figura 6.4. Superficies de logro.	132
Figura 6.5. Comparación cualitativa del enfoque UTH.	136
Figura 6.6. Comportamiento de la longitud de del vuelo (fl) durante las iteraciones de MOCCSA.	141
Figura 6.7. Comparación de gráficos de caja de los valores de IGD MOCCSA vs. MOCSA. ..	146
Figura 6.8. Comparación cualitativa, Pareto verdadero (azul) vs. mejor frente encontrado por cada algoritmo (rojo).	151
Figura 6.9. Superficies de logro de funciones de prueba bi-objetivo.	152

Índice de tablas

Tabla 3.1. Parámetros del EMO para ECM.....	49
Tabla 3.2. Resultados obtenidos por EMO aplicando la ECM sobre el conjunto de imágenes de prueba.....	49
Tabla 3.3. Resultados obtenidos después de aplicar el algoritmo EMO para ECM sobre las imágenes seleccionadas.....	52
Tabla 3.4. Comparación de los valores de media y de los algoritmos EMO, DE, PSO, HS, SSA y ABC aplicados sobre el subconjunto de imágenes de prueba usando ECM como función objetivo.....	54
Tabla 3.5. Comparaciones de los valores de PSNR, SSIM y FSIM generados por los algoritmos EMO, DE y PSO.....	55
Tabla 3.6. Comparación de los valores de PSNR,SSIM y FSIM obtenidos por los algoritmos HS, SSA, y ABC al ser aplicados en la segmentación del subconjunto de imágenes de prueba.....	55
Tabla 3.7. Valores p de la prueba de Wilcoxon comparando EMO vs. DE, EMO vs. PSO, EMO vs. HS, EMO vs. SSA y EMO vs. ABC.....	56
Tabla 3.8. Media del tiempo computacional de los algoritmos EMO, DE, PSO, HS, SSA y ABC para la segmentación de imágenes usando la ECM.....	57
Tabla 3.9. Comparación la evolución de los valores de aptitud de DE (línea roja), PSO (línea cian), HS (línea verde), SSA (línea magenta), ABC (línea negra) and EMO (línea azul).....	59
Tabla 4.1. Parámetros seleccionados para CSA, DE y HS.....	66
Tabla 4.2. Mejores umbrales encontrados por CSA, DE y HS.....	67
Tabla 4.3. Valores de la función objetivo de las distintas implementaciones de ECM para el primer conjunto de imágenes de prueba generales.....	68
Tabla 4.4. Comparación de los valores de PSNR obtenidos por CSA, DE y HS para entropía cruzada y el método de Otsu.....	69
Tabla 4.5. Comparación de los valores de SSIM y FSIM para el conjunto de imágenes generales de referencia.....	71
Tabla 4.6. Resultados visuales de la metodología ECM-CSA sobre el conjunto de imágenes de referencia general.....	72
Tabla 4.7. Resultados de ECM-CSA aplicado al conjunto de imágenes de RM cerebrales.....	74
Tabla 4.8. Resultados de calidad de las versiones de ECM aplicadas a imágenes de RM (El número en el nombre de las imágenes indica el valor del plano z de la RM).....	75
Tabla 4.9. Resultados cualitativos del algoritmo ECM-CSA aplicados a resonancias magnéticas cerebrales (nt=5).....	76
Tabla 4.10. Valores p obtenidos por la prueba de Wilcoxon entre CSA vs. DE y CSA vs. HS para el conjunto de imágenes generales de referencia.....	77
Tabla 4.11.. Valores p obtenidos por la prueba de Wilcoxon entre CSA vs. DE y CSA vs. HS para el conjunto de RM cerebrales.....	79
Tabla 4.12. Imágenes térmicas de prueba.....	86
Tabla 4.13. Parámetros de implementación.....	87
Tabla 4.14. Resultados estadísticos de ABC, DE, HS y FPA utilizando el método Otsu como función objetivo.....	88
Tabla 4.15. Resultados de calidad de la imagen segmentada para ABC y DE utilizando el método Otsu.....	88
Tabla 4.16. Resultados de calidad de la imagen segmentada para HS y FPA utilizando el método Otsu.....	89
Tabla 4.17. Resultados estadísticos de ABC, DE, HS y FPA utilizando la entropía de Kapur como función objetivo.....	89
Tabla 4.18. Resultados de calidad de la imagen segmentada para ABC y DE utilizando la entropía de Kapur.....	90
Tabla 4.19. Resultados de calidad de la imagen segmentada para HS y FPA utilizando la entropía de Kapur.....	90
Tabla 4.20. Resultados cualitativos utilizando la entropía de Kapur.....	91
Tabla 5.1. Umbrales obtenidos por los algoritmos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando el método de Otsu como función objetivo.....	102
Tabla 5.2. Media y σ de los valores de PSNR obtenidos por los algoritmos SCA, ALO, GA, PSO, SCA, RRA y ACOR usando Otsu como función objetivo.....	103

Tabla 5.3. Media y σ de los valores de SSIM obtenido por los algoritmos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando Otsu como función objetivo.	103
Tabla 5.4. Media y σ de los valores de FSIM obtenidos por los algoritmos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando Otsu como función objetivo.	104
Tabla 5.5. Imágenes segmentadas y su curva de energía umbralada usando el algoritmo ALO con la función objetivo propuesta por Otsu.	104
Tabla 5.6. Imágenes segmentadas y su curva de energía umbralada mediante el algoritmo SCA y la función objetivo de Otsu.	106
Tabla 5.7. Umbrales encontrados por cada algoritmo (SCA, ALO, GA, PSO, CSA, RRA y ACOR) usando el método de Kapur como función objetivo.	107
Tabla 5.8. Media y σ de los valores de PSNR obtenidos por los métodos ALO, SCA, GA, PSO, CSA, RRA y ACOR usando el método de Kapur.	108
Tabla 5.9. Media y σ de los valores de SSIM obtenidos por los métodos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando el método de Kapur.	109
Tabla 5.10. Media y σ de los valores de FSIM obtenidos por los métodos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando el método de Kapur.	109
Tabla 5.11. Imágenes segmentadas y su curva de energía umbralada usando el algoritmo ALO y el método de Kapur como función objetivo.	109
Tabla 5.12. Imágenes segmentadas y curva de energía umbralada por el algoritmo SCCA usando Kapur como función objetivo.	110
Tabla 5.13. Valores p de la prueba de Wilcoxon realizada sobre los valores de la función objetivo de Otsu.	112
Tabla 5.14. Valores p de la prueba de Wilcoxon realizada sobre los valores de la función objetivo de Kapur.	112
Tabla 6.1. Umbrales seleccionados por cada algoritmo.	128
Tabla 6.2. Comparación con enfoques anteriores.	128
Tabla 6.3. Valor de primero objetivo y desviación estándar.	129
Tabla 6.4. Valor del segundo objetivo y su desviación estándar.	129
Tabla 6.5. Hipervolumen y desviación estándar.	130
Tabla 6.6. Los mejores umbrales encontrados por cada MOEA.	132
Tabla 6.7. Valores de la relación señal/ruido pico (PSNR) para cada una de las desviaciones estándar de la técnica.	133
Tabla 6.8. Índice de Similitud Estructural (SSIM) y desviación estándar.	133
Tabla 6.9. Índice de similitud de las características (FSIM) y desviación estándar.	133
Tabla 6.10. Parámetros seleccionados.	142
Tabla 6.11. Conjunto de pruebas CEC 2009.	142
Tabla 6.12. Resultados estadísticos de la IGD para UF1 a UF10.	146
Tabla 6.13. Resultados estadísticos de SP para UF1 a UF10.	147
Tabla 6.14. Resultados estadísticos de la MS para UF1 a UF10.	148
Tabla 6.15. Resultados estadísticos del hipervolumen de UF1 a UF10.	149
Tabla 6.16. Resultados de la prueba estadística de Wilcoxon.	149

Resumen

Uno de los temas más tratados en la comunidad de procesamiento de imágenes es la segmentación, consistente en obtener estructuras subyacentes para facilitar su interpretación, por ejemplo, obtener bordes o agrupaciones de píxeles que forman regiones con alguna propiedad. Dado que se utiliza como un paso de preprocesamiento antes de las tareas de visión por computador de alto nivel, como el reconocimiento de objetos y la representación de imágenes, se han propuesto diversos enfoques para la segmentación, que se centran en la mejora de la calidad de los procesos aplicados para conseguir los mejores resultados posibles. Sin embargo, en muchos casos el coste computacional de estas técnicas puede ser elevado, limitando su uso.

En el área de la optimización global se han propuesto una gran cantidad de algoritmos metaheurísticos (AM) para resolver problemas complejos de ingeniería en un tiempo razonable. Los AMs son algoritmos de búsqueda estocásticos que utilizan reglas o heurísticas aplicables a cualquier problema para acelerar su convergencia a soluciones cercanas al óptimo. Es común observar que los AMs emulan procesos y comportamientos inspirados por mecanismos presentes en la naturaleza, como la evolución.

En la literatura se ha demostrado la factibilidad de realizar distintas tareas del procesamiento de imágenes, orientado hacia la segmentación, mediante el uso de AMs, como la umbralización de imágenes (UI). En general, los enfoques de UI realizan la separación de los píxeles de una imagen considerando determinados valores de intensidad (umbrales) que particionan el histograma de la imagen en un número finito de clases según los valores de los píxeles con respecto a los umbrales. Así, el problema se reduce a realizar una búsqueda de los mejores valores umbrales para una imagen de acuerdo con su histograma. La calidad de los umbrales mediante AM se determina mediante el uso de criterios no paramétricos, donde un AM explora el espacio de soluciones hasta encontrar umbrales óptimos.

Tras el correspondiente análisis de diversos métodos existentes en la literatura se han encontrado deficiencias, habiéndose propuesto los métodos que se indican a continuación para su resolución, que a su vez constituyen la base de la investigación desarrollada en la tesis y la contribución a la comunidad científica, concretándose como sigue en los siguientes aspectos: a) mejora de la calidad en la segmentación usando la entropía cruzada mínima (divergencia) como criterio a optimizar; b) estudio del desempeño de los AMs en aplicaciones, tales como resonancias magnéticas e imágenes térmicas; c) incorporación de información contextual para incrementar la mejora de la umbralización, como técnica específica de segmentación; y d) propuesta de un enfoque multiobjetivo en optimización para determinar el número de umbrales automáticamente, cuya finalidad es la segmentación.

En el primer aspecto se busca mejorar la calidad de las imágenes segmentadas aplicando AMs mediante el uso criterios de umbralización investigados recientemente con resultados satisfactorios. La mayoría de los trabajos relacionados con la umbralización de imágenes utiliza como criterio de calidad la varianza entre clases o la maximización de la entropía a pesar de la existencia de criterios más eficaces como la divergencia, también llamada entropía cruzada mínima. Para ello se evalúa el desempeño de la entropía cruzada usando distintos AMs sobre imágenes de prueba ampliamente utilizadas en la literatura relacionada. Además, se analizan los resultados de acuerdo con la calidad de la segmentación y la convergencia de los algoritmos que se proponen.

El segundo aspecto estudia el desempeño en de AMs en dos casos de aplicación concretos: resonancias magnéticas cerebrales e imágenes térmicas. Cada una de estas aplicaciones requiere características especiales que deben ser consideradas en el proceso de segmentación, dada la naturaleza de las imágenes.

El tercer aspecto tiene como objetivo lograr la incorporación de información contextual en el proceso de umbralización, esto es, considerar información correspondiente a la vecindad de cada píxel para que el resultado de la umbralización sea más eficiente. Esto se logra usando la curva de energía de la imagen como sustituto del histograma. La curva de energía tiene propiedades similares a las del histograma, por lo que es posible reemplazarlos. El uso de la curva contribuye a realizar una mejor segmentación en imágenes con ruido. Así, se busca implementar AMs desarrollados recientemente y clásicos para establecer un marco de comparación.

Finalmente, dado que en la literatura la gran mayoría de contribuciones relacionadas con AMs y procesamiento de imágenes se realizan como problemas de optimización de un solo objetivo, se ha decidido proponer un enfoque multiobjetivo que permite realizar segmentación de imágenes y simultáneamente determinar la mejor cantidad de umbrales y su posición óptima

a lo largo del histograma de la imagen. De esta forma se puede resolver uno de los problemas más importantes inherentes a la umbralización, la determinación *a priori* de la cantidad de umbrales.

Las estrategias propuestas se presentan como alternativas para dar solución a los temas seleccionados en el área de la segmentación de imágenes. La evidencia experimental y análisis comparativos realizados de forma independiente permiten comprobar su desempeño en considerando elementos como su precisión y robustez. Además, se comprueba que el uso de los AM es una alternativa viable en una gran cantidad de variantes de la umbralización de imágenes, lo que contribuye positivamente al estado del arte en el tema investigado.

Palabras clave: algoritmos metaheurísticos, optimización, segmentación, umbralización.

Abstract

One of the most discussed topics in the image processing community is segmentation. Since it is used as a pre-processing step prior to high-level computer vision tasks, such as object recognition and image representation, various approaches to image segmentation have been proposed that focus on improving the quality of the resulting segmentation. However, in many cases the computational cost of these techniques can be high.

In the area of global optimization, they have proposed a large number of metaheuristic (MA) algorithms to solve complex engineering problems in a reasonable amount of time. MAs are stochastic search algorithms that use rules or heuristics applicable to any problem to accelerate their convergence to near-optimal solutions. It is common to observe that MAs emulate processes and behaviors inspired by mechanisms present in nature, such as evolution.

In the literature, the feasibility of performing different image processing tasks using MAs, such as image thresholding (TH), has been demonstrated. In general, TH approaches perform the separation of the pixels of an image according to the threshold values that partition the image histogram into a finite number of classes. Thus, the problem is reduced to finding the best thresholds for an image based on the histogram. The quality of the thresholds is determined by using non-parametric criteria, where an MA explores the solution space until optimum thresholds are found.

After the corresponding analysis of different existing methods in the bibliography, deficiencies have been found, and the following methods have been proposed for their resolution, which in turn constitute the basis of the research developed in the thesis: The improvement of the quality of segmentation using divergence as a criterion to be optimized, the incorporation of contextual information to the threshold, the study of the performance of MAs

in applications such as magnetic resonances and thermal images and the proposal of a multi-objective approach to determine the number of thresholds automatically. Thus, the following paragraphs detail each of the selected topics.

The first theme seeks to improve the quality of segmented images using MAs by utilizing more recent threshold criteria. Most of the work related to the threshold of images uses as a quality criterion the variance between classes or the maximization of entropy despite the existence of more effective criteria such as divergence, also called minimal cross entropy. For this purpose, the performance of cross entropy is evaluated using different MAs on a set of test images. In addition, the results are analyzed according to the quality of the segmentation and the convergence of the algorithms.

The second topic studies the performance of MAs in two specific application cases: brain MRIs and thermal imaging. Each of these applications involves special characteristics that must be considered in the segmentation process given the nature of the images.

The third theme aims to achieve the incorporation of contextual information in the thresholding process, i.e. to consider information corresponding to the vicinity of each pixel for its thresholding. This is accomplished by using the energy curve in the image as a substitute for the histogram. The energy curve has properties similar to those of the histogram, so it is possible to replace them. The use of the curve contributes to better segmentation in noisy images. Thus, it is sought to implement recent and classic MAs to establish a framework for comparison.

Finally, given that in the literature the vast majority of contributions related to MAs and image processing are made as single objective optimization problems, it has been decided to propose a multi-objective approach that allows for image segmentation and simultaneously determine the best number of thresholds and their optimal position along the image histogram. In this way, one of the most important problems of the threshold, the a priori determination of the number of thresholds, can be solved.

The proposed strategies are presented as alternatives to solve the selected issues in the area of image segmentation. The experimental evidence and comparative analyses carried out in an independent way allow to verify its performance considering elements such as its precision and robustness. In addition, it is confirmed that the use of AM is a viable alternative in many variants of the thresholding of images, which contributes positively to the state of the art in the subject under investigation.

Keywords: metaheuristic algorithms, optimization, segmentation, thresholding.

1 Introducción

Tal y como refleja el título de la memoria, el desarrollo de la investigación se enfoca en la segmentación de imágenes. La segmentación es un concepto cuyo objetivo consiste en obtener estructuras subyacentes para facilitar su interpretación a alto nivel, por ejemplo, obtener bordes o agrupaciones de píxeles que forman regiones con alguna propiedad común. La optimización (mono y multi-objetivo) constituye un pilar esencial en los procesos de optimización que se plantean. Dentro de la segmentación se encuadran los procesos conocidos con umbralización, cuyo objetivo final consiste en separar agrupaciones de píxeles mediante la selección óptima de valores de intensidad específicos (umbrales), donde la optimización está llamada a jugar un papel relevante.

En las últimas décadas, la creciente disponibilidad de cámaras digitales ha fomentado el desarrollo de sistemas basados en la visión para resolver problemas en áreas tan diversas como la medicina, topología, agricultura o la vigilancia, entre otras muchas (Osuna-Enciso, Cuevas, & Sossa, 2013; Zaitoun & Aqel, 2015; Y. J. Zhang, 1996). A pesar de las distintas naturalezas de las aplicaciones, la mayoría de los sistemas requieren la segmentación de los objetos presentes en la imagen. Como resultado, la segmentación de imágenes se ha convertido en un tema de investigación atractivo.

Uno de los métodos de segmentación más utilizados, debido a su independencia con respecto del tamaño de la imagen y su facilidad de implementación, es la umbralización de imágenes (UI) (Gonzalez & Woods, 1992). En general, los enfoques de UI realizan una separación de los píxeles en la imagen de acuerdo con su histograma, el valor de un píxel dado y uno o más valores de intensidad, conocidos como umbrales (Sahoo, Soltani, & Wong, 1988). Originalmente el UI fue desarrollado para la separación del primer plano y fondo en una imagen

recibiendo el nombre de binarización. Posteriormente, la formulación se amplió para incorporar más de dos clases, lo que condujo a la definición de la umbralización multinivel (UM) (Kumar, Kumar, Sharma, & Pant, 2013). Ambos enfoques consideran como base conceptual la frecuencia de ocurrencia de cada valor de intensidad en la imagen para generar un histograma; luego, los valores umbral dividen secciones del histograma para separar diferentes clases. Así, el problema de UM se puede resumir como la búsqueda de los mejores valores umbral que segmentan una imagen.

En el caso de la binarización, la búsqueda del mejor valor de umbral puede ser realizada de forma exhaustiva; esto es, probar todos los umbrales posibles hasta encontrar el valor de umbral que separe la imagen de la mejor manera posible. Sin embargo, el UM no puede ser tratado exhaustivamente ya que con cada valor de umbral que se agrega a la imagen, la complejidad de la búsqueda crece exponencialmente (Akay & Bahriye, 2013; Hammouche, Diaf, & Siarry, 2010; Liao, Chen, & Chung, 2001).

Para evitar realizar una búsqueda con todas las posibles combinaciones de umbrales, en la literatura se han adoptado técnicas de optimización que permiten encontrar la solución a problemas complejos mediante la minimización o maximización de un criterio de calidad. De esta forma es posible reducir el tiempo que llevaría encontrar una solución óptima. Dentro de la familia de algoritmos de optimización es posible encontrar una gran variedad de enfoques que tienen como finalidad encontrar la solución óptima para un problema dado. Entre éstas es posible encontrar enfoques pertenecientes al grupo de la inteligencia colectiva (Blum & Merkle, 2008).

Los algoritmos evolutivos, *evolutionary algorithms* (EA), consideran el uso de agentes capaces de explorar eficientemente algunas de las posibles soluciones del problema que se analiza. Los EAs están compuestos por poblaciones de soluciones que se desplazan o evolucionan durante un proceso iterativo hasta que se cumple una condición de paro. Los EAs establecen reglas simples que permite generar en conjunto comportamientos colectivos inteligentes. Dado que en la literatura no existe una taxonomía universalmente aceptada, estos algoritmos suelen recibir también el nombre de algoritmos poblacionales (AP) perteneciendo a la familia de los metaheurísticos (AM) (X. Yang, 2010).

1.1 Antecedentes

En ingeniería, prácticamente todo puede ser optimizado. La optimización puede incluirse en diferentes campos, por ejemplo, control de calidad, diseño, vigilancia, etc. El objetivo es encontrar los mejores valores que resuelvan el problema en cuestión. Por ejemplo, minimizar el coste de producción de un producto o eficiencia energética. Los métodos de

optimización se pueden dividir en dos grandes grupos: deterministas y estocásticos. Las técnicas deterministas incluyen enfoques clásicos y la mayoría de ellos utilizan operaciones de gradiente. Por otro lado, los enfoques estocásticos incluyen el uso de reglas (heurísticas) que imitan el comportamiento de diferentes procesos en la naturaleza. De forma simple podemos definir una heurística como un conjunto de reglas que pueden ayudar en la solución de un problema específico, mientras que las herramientas metaheurísticas son aquellas diseñadas para funcionar en cualquier tipo de problema. Recientemente, varios algoritmos metaheurísticos han sido propuestos con resultados interesantes (Askarzadeh, 2016; Fausto et al., 2017; Mirjalili, et al., 2017). Tales enfoques utilizan nuestra comprensión científica de los sistemas biológicos, naturales o sociales, que en algún nivel de abstracción pueden ser representados como procesos de optimización.

Los EAs son algoritmos de optimización libres de derivada que buscan soluciones óptimas utilizando vectores de diseño (soluciones) que evolucionan a lo largo de un espacio de búsqueda factible de acuerdo con un conjunto de reglas y operadores. Por lo general, las reglas de los EAs se inspiran en comportamientos observados en la naturaleza, como la evolución, de ahí su nombre. En este sentido, los EAs son adecuados para problemas muy complejos en los que las búsquedas exhaustivas no son prácticas y los métodos clásicos pueden fracasar. Sin embargo, el desempeño de los EAs está fuertemente ligado al equilibrio entre las operaciones de exploración y explotación. Ambas tareas son contradictorias y deben ser consideradas en cualquier aplicación (Talbi, 2002). Una alta explotación puede conducir a un estancamiento local, mientras que una exploración intensiva podría impedir la convergencia del algoritmo. Estas diferencias han llevado a la formulación de una gran cantidad de métodos metaheurísticos, donde distintos comportamientos son emulados con el objetivo de obtener mejores algoritmos de optimización. Además, David Wolpert en una serie de pruebas lógicas propone el teorema *No-Free-Lunch* para la optimización (Wolpert & Macready, 1997), donde afirma que ningún algoritmo de optimización funcionará bien en todos y cada uno de los problemas. Por lo tanto, la evaluación de un determinado EA sobre una tarea específica no podía reflejar con precisión las capacidades de dicho método.

Dentro de la familia de algoritmos metaheurísticos, los métodos que más han impactado el área y son de interés para la investigación que se refleja en esta tesis son los algoritmos genéticos que emulan la evolución mediante el paso de genes, *genetic algorithms* (GA) (Goldberg & Holland, 1988), el comportamiento social de los bancos de peces y bandadas de aves en algoritmo de optimización de enjambres de partículas, *particle swarm optimization* (PSO) (Kennedy & Eberhart, 1995), la emulación de la evolución, en su versión diferencial, *differential evolution* (DE) (Storn & Price, 1997), el proceso de improvisación de armonías que ocurre por ejemplo cuando un músico busca un estado o satisfacción musical más favorable,

harmony search (HS) (Geem, Kim, & Loganathan, 2001) y la simulación del fenómeno del electromagnetismo como el algoritmo basado en el electromagnetismo, *electromagnetism-like optimzation* (EMO) (Ş. I. Birbil & Fang, 2003).

Además, es posible encontrar algoritmos propuestos durante esta década que continúan explorando el uso de metáforas que inspiran el funcionamiento de cada algoritmo. Entre los más importantes destaca el comportamiento cooperativo de las colonias de abejas, como la técnica de colonias artificiales de abejas, *artificial bee colony* (ABC) (Dervis Karaboga & Basturk, 2007), la emulación del comportamiento del murciélago como el método *bat algorithm* (BA) (Yang, 2010), el comportamiento de apareamiento de las luciérnagas, *firefly* (FFA) (Yang, 2010), la simulación del comportamiento animal en un grupo como el *collective animal behavior* (CAB) (Cuevas, González, Zaldivar, Pérez-Cisneros, & García, 2012), la optimización basada en la araña social *social spider algorithm* (SSA) (Yu & Li, 2015), el algoritmo de la hormiga león *ant lion optimizer* (ALO) (Mirjalili, 2015), algoritmo de tallos *runner-root algorithm* (RRA) (Merrikh-Bayat, 2015), el comportamiento de la araña social como la *social spider optimization* (SSO) (Cuevas, Díaz Cortés, & Oliva Navarro, 2016), el algoritmo basado en seno y coseno *sine cosine algorithm* (SCA) (Mirjalili, 2016b) y por último la búsqueda del cuervo *crow search algorithm* (CSA) (Askarzadeh, 2016).

El inconveniente del uso de EAs es la necesidad de adaptar el problema a resolver desde el punto de vista de la optimización. En otras palabras, es necesario definir la función objetivo o función de aptitud. En el caso de la segmentación de imágenes es posible encontrar una gran variedad de implementaciones basadas en EAs con distintos grados de éxito, algunas de las cuales se exponen en esta memoria.

1.2 Planteamiento del problema

En la literatura se ha demostrado la factibilidad de realizar distintas tareas relativas al procesamiento de imágenes mediante el uso de EAs, incluida la segmentación. No obstante, tras un análisis de la bibliografía existente en la comunidad científica se han encontrado deficiencias, algunas de las cuales son precisamente las que se han abordado en este trabajo para proceder a su resolución. Específicamente se ha decidido tratar cuatro deficiencias mediante los siguientes temas: mejorar la calidad de la segmentación, incorporar información contextual, proponer un enfoque multiobjetivo y estudiar el desempeño de EAs en aplicaciones específicas (imágenes térmicas y de resonancia magnética). Así, en los siguientes párrafos se detalla cada uno de los temas seleccionados.

En el primer tema se busca mejorar la calidad de las imágenes segmentadas usando EAs mediante el uso de criterios de umbralización recientes. La mayoría de los trabajos relacionados

con la umbralización de imágenes utiliza como criterio de calidad la varianza entre clases o la maximización de la entropía, ello a pesar de la existencia de criterios más eficaces como la divergencia, también llamada entropía cruzada. Para ello, en este trabajo, se evalúa el desempeño de la entropía cruzada usando distintos EAs sobre un conjunto de imágenes de prueba. Además, se analiza la evidencia reportada por los resultados de acuerdo con la calidad de la segmentación y la convergencia de los algoritmos.

El segundo tema estudia el desempeño de EAs en dos casos de aplicación concretos: resonancias magnéticas cerebrales e imágenes térmicas. Cada una de estas aplicaciones supone la existencia de características especiales que deben ser consideradas en el proceso de segmentación, dada la naturaleza de las imágenes.

El tercer tema tiene como objetivo lograr la incorporación de información contextual en el proceso de umbralización, esto es, considerar información correspondiente a la vecindad de cada píxel para su umbralización. Esto se logra usando la curva de energía de la imagen como sustituto del histograma. La curva de energía tiene propiedades similares a las del histograma, por lo que es posible reemplazarlos. El uso de la curva contribuye a realizar una mejor segmentación en imágenes con ruido. Así, se busca implementar EAs recientes y clásicos para establecer un marco de comparación.

Finalmente, dado que en la literatura la gran mayoría de contribuciones relacionadas con EAs y procesamiento de imágenes se realizan como problemas de optimización de un solo objetivo, se ha decidido proponer un enfoque multiobjetivo que permite realizar segmentación de imágenes y simultáneamente determinar la mejor cantidad de umbrales y su posición óptima a lo largo del histograma de la imagen. De esta forma se puede resolver uno de los problemas más importantes del UM, la determinación *a priori* del número de umbrales.

1.3 Objetivos

Después de realizar un análisis de los antecedentes y las problemáticas a desarrollar, se han definido los objetivos principales de esta investigación. Para mayor claridad se presentan como objetivos específicos y generales.

1.3.1 Objetivos generales

1. Utilizar algoritmos evolutivos para la solución del problema de segmentación de imágenes.
2. Desarrollar implementaciones alternativas a las técnicas de segmentación tradicionales para tipos de imágenes específicas (resonancias magnéticas, imágenes térmicas).

1.3.2 Objetivos específicos

1. Analizar la bibliografía existente en los temas de EAs y segmentación de imágenes.
2. Presentar una nueva alternativa de UM para imágenes digitales basado en el algoritmo EMO con el fin de minimizar la entropía cruzada.
3. Implementar el uso de la curva de energía para el problema de UM usando métodos tradicionales como el de Otsu (1979) y Kapur et al. (1985) en algoritmos metaheurísticos clásicos y recientes.
4. Proponer nuevo enfoque multiobjetivo para la segmentación de imágenes digitales.
5. Crear una mejora del algoritmo CSA que le permita funcionar en problemas multiobjetivo. Para favorecer la diversidad en las soluciones se usarán atractores caóticos para determinar parámetros del algoritmo.
6. Usando el algoritmo CSA junto con la entropía cruzada, proponer un método para segmentar imágenes de resonancias magnéticas cerebrales.
7. Evaluar el desempeño de algoritmos de optimización para la segmentación de imágenes térmicas.
8. Difundir los resultados encontrados por la investigación publicando artículos científicos u otros medios.

1.4 Metodología

La investigación descrita en la presente memoria de tesis se ha realizado de acuerdo con las pautas establecidas por el método científico. Como primer paso se ha realizado una búsqueda bibliográfica en todas las fuentes académicas disponibles, utilizando bases de datos internacionales a las cuales la Universidad Complutense de Madrid tiene acceso. Las áreas previamente mencionadas han sido estudiadas para establecer sus antecedentes. Además, debe hacerse notar que este trabajo de investigación fue desarrollado en conjunto con la Universidad de Guadalajara, concretamente con el Centro Universitario de Ciencias Exactas e Ingenierías en Guadalajara México. Posteriormente se desarrollaron las etapas de trabajo que se indican a continuación:

1. Definición de la problemática que se desea abordar mediante el uso de algoritmos metaheurísticos, siendo éstos el objeto de la investigación.
2. Planteamiento del alcance y objetivos de la investigación
3. Diseño y desarrollo de las metodologías.
4. Realización de experimentos y obtención de resultados experimentales.
5. Análisis e interpretación de los datos.
6. Conclusión general.
7. Planteamiento de trabajo a futuro.

Cada problema abordado en esta investigación tiene un conjunto independiente de experimentos que resultan apropiados para evaluar las técnicas propuestas. El proceso completo incluye la adquisición de datos, procesamiento de la información, desarrollo y codificación de las distintas variantes de algoritmos de optimización metaheurísticos seleccionados, obtención de resultados y finalmente su validación.

1.5 Aportaciones de la investigación

Durante el desarrollo de esta tesis doctoral se ha logrado dar solución a los problemas previamente planteados, expresados en los objetivos indicados. Con el fin de presentar ante la comunidad científica los resultados experimentales de las distintas problemáticas resueltas, se han realizado las siguientes publicaciones: a) cinco artículos en revistas indexadas en el Journal Citation Report; b) un capítulo de libro y c) dos ponencias en congresos.

1.5.1 Artículos publicados en revistas indexadas en el Journal Citation Reports (JCR)

Autores: Diego Oliva, **Salvador Hinojosa**, Valentín Osuna-Enciso, Erik Cuevas, Marco Pérez-Cisneros, Gildardo Sanchez-Ante
Título: Image segmentation by minimum cross entropy using evolutionary methods
Revista: Soft Computing (**Q2 Computer science, artificial intelligence IF 2.367**)
Pag. (año): 1-20, (2017)

Como se ha indicado previamente, la segmentación de imágenes digitales es uno de los pasos más importantes en los sistemas de visión por computador. Varias técnicas de segmentación han sido publicadas donde algunas de ellas usan operadores complejos. Las técnicas basadas en la umbralización suelen ser fáciles de implementar. Ahí, el problema es determinar correctamente los mejores umbrales que dividen los valores de intensidad de los píxeles. Un método interesante para elegir los mejores umbrales es la entropía cruzada mínima (ECM) (Li & Lee, 1993), la cual ha demostrado excelentes resultados en tareas de binarización. Sin embargo, la extensión del problema de segmentación para incluir múltiples umbrales incrementa significativamente el esfuerzo computacional requerido para encontrar valores de umbrales óptimos. Cada nuevo umbral incrementa la complejidad de la formulación del problema. Los métodos clásicos para segmentar imágenes realizan búsquedas exhaustivas, mientras que algunos métodos nuevos aprovechan las heurísticas para reducir el espacio de búsqueda. Los algoritmos evolutivos usan heurísticas para optimizar algún griterío sobre un número finito de iteraciones. La correcta selección de un algoritmo para optimizar el problema de la ECM impacta directamente en el desempeño del método. El algoritmo *electromagnetism-like optimization* (EMO) (Birbil & Fang, 2003) es una técnica evolutiva, la cual simula el mecanismo de atracción-repulsión entre cargas para evolucionar la población. Tal técnica requiere sólo un pequeño número de evaluaciones para encontrar un óptimo. En este artículo se propone el uso del algoritmo EMO para buscar umbrales óptimos minimizando la función de la entropía cruzada mientras se reduce el número de iteraciones y acceso a funciones. Esta propuesta es evaluada en un conjunto de imágenes de prueba para demostrar que es capaz de

mejorar la velocidad de convergencia. Los resultados descritos por este artículo están plasmados en el capítulo tres.

Autores: Diego Oliva, **Salvador Hinojosa**, Erik Cuevas, Gonzalo Pajares, Omar Avalos, Jorge Gálvez
Título: Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm
Revista: Expert Systems with Applications (**Q1 Computer science, artificial intelligence. IF 3.768**)
Vol, pag. (año): 79, 164-180, (2017)

El proceso de segmentación es considerado parte central en los sistemas de procesamiento de imágenes, dada su alta influencia en procesos posteriores. En años recientes, la segmentación de imágenes de resonancias magnéticas (RM) ha atraído la atención de la comunidad científica con el objetivo de ayudar en el diagnóstico de numerosas enfermedades cerebrales. De entre las muchas técnicas existentes, la umbralización representa uno de los métodos más populares para la segmentación de imágenes. Actualmente, una gran cantidad de contribuciones se han hecho en la literatura, donde los valores de umbrales son obtenidos mediante la optimización de algún criterio relevante como la entropía de Kapur et al. (1985). Sin embargo, la mayoría de esos métodos conllevan un alto coste computacional dado que utilizan búsquedas exhaustivas para encontrar los valores de umbral óptimos. Este artículo presenta un método general para la segmentación de imágenes de RM. Para determinar los valores de umbrales, el algoritmo propuesto usa el algoritmo evolutivo de referencia llamado *Crow Search Algorithm* (CSA) (Askarzadeh, 2016) el cual está basado en el comportamiento de bandadas de cuervos. A diferencia de otras técnicas de optimización usadas para segmentar, CSA presenta un mejor desempeño, evitando fallas críticas como la convergencia prematura a soluciones subóptimas. A pesar de que el algoritmo propuesto puede funcionar como un algoritmo de segmentación genérico, sus características le permiten obtener resultados excelentes en la segmentación automática de RMs. En tales circunstancias, el enfoque propuesto ha sido evaluado usando dos conjuntos de imágenes, el primero está compuesto de imágenes genéricas usadas en la literatura del procesamiento de imágenes, mientras que el segundo conjunto pertenece a RMs. Los resultados experimentales, estadísticamente validados, demostraron que la técnica propuesta obtiene mejores resultados en términos de calidad y consistencia. El contenido de este artículo es ampliado en el capítulo cuatro.

Autores: **Salvador Hinojosa**, Omar Avalos, Diego Oliva, Erik Cuevas, Gonzalo Pajares, Daniel Zaldivar, Jorge Gálvez
Título: Unassisted Thresholding based on Multi-Objective Evolutionary Algorithms
Revista: Knowledge-Based Systems (**Q1 Computer science, artificial intelligence. IF 4.369**)
Vol, Pag. (año): 159, 221-232, (2018)

El sexto capítulo se dedica a presentar los enfoques multiobjetivo que pueden ser aplicados a la segmentación. La primera sección del capítulo sexto muestra las contribuciones realizadas en este artículo. Los algoritmos evolutivos multiobjetivo (MOEAs) resuelven problemas en los que hay dos o más objetivos en conflicto. Para lograrlo, los MOEAs incorporan estrategias para determinar compromisos óptimos entre cada una de las funciones objetivo. En este trabajo, se propone una metodología de umbralización de imagen sin asistencia (UTH) basada en MOEAs. UTH se aprovecha de los mecanismos de compromiso presentes en

los MOEAs para realizar la umbralización de imágenes donde se determina simultáneamente el número de umbrales requeridos para segmentar cada imagen y la mejor ubicación de cada umbral a lo largo del histograma. El problema de la umbralización de imágenes se trata comúnmente como la búsqueda de los mejores umbrales posibles capaces de dividir una imagen dada en un número finito de clases homogéneas. Este enfoque requiere la asistencia de un diseñador para determinar el número de valores umbral que segmentarán correctamente la imagen. Sin embargo, como las imágenes pueden variar significativamente, la definición de un número óptimo de umbrales debe realizarse para cada imagen. Así, una metodología capaz de determinar tanto el número de umbrales como la mejor ubicación de cada valor contribuye al diseño de un esquema general de segmentación. En el enfoque propuesto, UTH redefine el problema de la umbralización como una tarea multiobjetivo con dos objetivos en conflicto. El primer objetivo es la calidad de la imagen segmentada, y se calcula como un criterio no paramétrico para evaluar los puntos de umbral de los candidatos. El segundo objetivo es el número normalizado de puntos de umbral. Dado que el número de umbrales no es fijo, se utiliza una partícula que codifica los umbrales con longitud variable. La estrategia de UTH está acoplada con tres MOEAs, *Non-dominated Sorting Genetic Algorithm III* (NSGA-III), *Pareto Envelope-based Selection Algorithm II* (PESA-II) y *Multi-Objective Particle Swarm Optimization* (MOPSO) usando como criterio no paramétrico la entropía cruzada. De acuerdo con los resultados, la formulación UTH-NSGA-III supera a UTH-PESA-II y UTH-MOPSO en cuanto a la convergencia y calidad de la imagen resultante.

Autores: **Salvador Hinojosa**, Diego Oliva, Erik Cuevas, Gonzalo Pajares, Omar Avalos, Jorge Gálvez
Título: Improving multi-criterion optimization with chaos: a novel Multi-Objective Chaotic Crow Search Algorithm
Revista: Neural Computing and Applications (**Q1 Computer science, artificial intelligence. IF. 4.213**)
Vol, Pag. (año): 29 (8), 319-335, 2017

La segunda sección del capítulo seis presenta la investigación, reflejada en este artículo, que llevó a la presentación de dos algoritmos multiobjetivo, *Multi-Objective Crow Search Algorithm* (MOCSA) y una versión mejorada llamada *Multi-Objective Chaotic Crow Search Algorithm* (MOCCSA). Ambos métodos están basados en una versión mejorada del algoritmo *Crow Search Algorithm* (CSA). Los cuervos son animales inteligentes que tienen estrategias interesantes para proteger sus escondites de comida. Este comportamiento atractivo es llevado a un enfoque multiobjetivo. MOCCSA usa criterios basados en caos para favorecer la diversidad de las soluciones. Para determinar si el desempeño de la propuesta (MOCCSA) mejorada con caos es significativa, un análisis estadístico entre la propuesta y su contraparte sin caos (MOCSA) indica que los resultados obtenidos por ambos algoritmos son significativamente distintos entre sí. El desempeño de MOCCSA es evaluado con un conjunto de funciones de prueba estándar y los resultados son contrastados contra dos algoritmos ampliamente conocidos, MODA y MOPSO (Coello, 2002; Mirjalili, 2016a). Ambos resultados cuantitativos y cualitativos muestran que el algoritmo propuesto es competitivo.

Autores	Diego Oliva, Salvador Hinojosa , Mohamed Abd El Aziz, Noé Ortega-Sánchez
Título	Context Based Image Segmentation using Antlion Optimization and Sine Cosine Algorithm
Revista:	Multimedia Tools and Applications (Q2 Computer science, theory and methods IF 1.541)
Vol, Pag. (año):	77, 25761-25797, (2018)

La umbralización multinivel (UM) es uno de los métodos más utilizados para realizar segmentación de imágenes. Sin embargo, como la mayoría de los métodos se basan en el histograma de la imagen para realizar la segmentación, los métodos UM sólo consideran la frecuencia de aparición de cierto nivel de intensidad despreciando toda la información espacial. En el capítulo cinco se presenta la investigación realizada en este artículo para incluir información contextual en el proceso de umbralización. La información espacial puede ayudar a mejorar la calidad de las imágenes segmentadas dado que considera no solo el valor de la intensidad del píxel, también su vecindad. La curva de energía fue diseñada para llevar información espacial a una curva con las mismas propiedades que el histograma. En este artículo se evalúan dos algoritmos evolutivos recientemente publicados en la realización de UM con dos criterios clásicos de umbralización sobre la curva de energía. Los EAs utilizados son el *Antlion Optimizer* (ALO) y el *Sine Cosine Algorithm* (SCA) (Mirjalili, 2015, 2016b). Los métodos propuestos son evaluados intensivamente en términos de calidad y se presenta un análisis estadístico para comparar los resultados de estos algoritmos contra enfoques tradicionales. La evidencia experimental motiva a recomendar el uso de ALO para problemas de UM mientras que el SCA no ofrece evidencia de superioridad frente a métodos anteriores.

1.5.2 Capítulos de libros

Autores:	Salvador Hinojosa , Gonzalo Pajares, Erik Cuevas, Noé Ortega-Sanchez
Título:	Thermal Image Segmentation Using Evolutionary Computation Techniques
Libro:	Advances in Soft Computing and Machine Learning in Image Processing
Editorial:	Springer
Pag. (año):	63-88, 2018

Este trabajo analiza el desempeño de tres algoritmos metaheurísticos (AM) aplicados a la segmentación de imágenes térmicas, en inglés *forward looking infrared* (FLIR). Las imágenes térmicas FLIR representan un desafío para técnicas de procesamiento de imágenes, ya que los dispositivos de captura por lo general producen imágenes de baja resolución propensas al ruido y con bordes poco nítidos. AMs tradicionales como el *artificial bee colony* (ABC) (Karaboga, 2005), *differential evolution* (DE) (Storn & Price, 1997), *harmony search* (HS) (Geem, et al., 2001) y el recientemente publicado *flower pollination algorithm* (FPA) (Yang, et al., 2014) son implementados y evaluados usando como función objetivo el método de Otsu (1979), y la entropía de Kapur et al. (1985). La comparación presta especial atención a la calidad de las imágenes segmentadas mediante la evaluación de tres métricas específicas, *peak-to-signal noise ratio* (PSNR), *structural similarity index* (SSIM) y *feature similarity index*

(FSIM). Los resultados obtenidos en la segmentación de imágenes FLIR pueden ser consultados en la segunda sección del capítulo cuatro.

1.5.3 Congreso

Autores: **Salvador Hinojosa**, Diego Oliva, Erik Cuevas, Marco Pérez-Cisneros, Gonzalo Pajares
Título: Real-time video thresholding using evolutionary techniques and cross entropy
Conferencia: The IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)
Año: 2018

Los algoritmos evolutivos (EAs) están presentes en la mayoría de las áreas de la ciencia y la ingeniería donde surgen problemas difíciles. Sin embargo, los EAs se aplican a menudo a problemas de diseño en los que la velocidad de procesamiento no es un factor crucial. Esta tendencia ha llevado a que los EAs sean excluidos de las aplicaciones en tiempo real debido a su naturaleza iterativa. El procesamiento de imágenes se ha beneficiado de los EAs en muchas aplicaciones fuera de línea, pero se ha investigado poco sobre problemas de procesamiento de imágenes en tiempo real. En este trabajo se presenta la evaluación de los EAs aplicados a la umbralización de un flujo de imágenes en tiempo real. Los resultados indican que la evolución diferencial (DE) puede ser modificada para lograr un rendimiento en tiempo real en una implementación de un solo núcleo sin ninguna forma de paralelización. Estas circunstancias indican que el rendimiento puede mejorarse aún más con implementaciones multinúcleo o con la paralelización en la GPU. El trabajo presentado por este artículo se incluye como inicio del trabajo futuro, donde se plantea incrementar la velocidad de la umbralización mediante la incorporación de algoritmos metaheurísticos paralelizables.

Autores: Diego Oliva, **Salvador Hinojosa**, MV Demeshko
Título: Engineering applications of metaheuristics: an introduction
Conferencia: Journal of Physics: Conference Series
Año: 2017

Los algoritmos metaheurísticos son herramientas importantes que en los últimos años se han utilizado ampliamente en varios campos. En ingeniería, hay una gran cantidad de problemas que pueden ser resueltos desde un punto de vista de optimización. Este artículo es una introducción a cómo la metaheurística puede ser utilizada para resolver problemas complejos de ingeniería. Su uso produce resultados precisos en problemas que son computacionalmente costosos. Los resultados experimentales apoyan el rendimiento obtenido por los algoritmos seleccionados en problemas tan específicos como el diseño de filtros digitales, procesamiento de imágenes y diseño de celdas solares. El contenido de este artículo ayuda a sustentar el capítulo tres de esta memoria.

1.6 Descripción de los capítulos

Esta tesis se divide en siete capítulos para abordar las problemáticas descritas. El orden de los capítulos sigue el desarrollo temático de esta investigación.

- Capítulo 1: establece el contexto básico en el cual se desarrolla la investigación de esta tesis. Además, se puntualizan los objetivos y alcances de esta investigación.
- Capítulo 2: presenta los conceptos que dan sustento a este trabajo. Se incluyen técnicas y conceptos teóricos relevantes. Se presta especial atención al tema de optimización y umbralización de imágenes. Además, se establecen las herramientas que serán usadas para comparar el desempeño de las distintas implementaciones realizadas para dar respuesta a las problemáticas planteadas en la sección 1.3.
- Capítulo 3: describe cómo se ha mejorado la calidad de la umbralización utilizando la formulación de la entropía cruzada mínima.
- Capítulo 4: analiza el desempeño de distintas técnicas de umbralización aplicado a imágenes especiales. Se analizan dos casos, el de resonancias magnéticas y el de imágenes térmicas. En cada caso se abordan las problemáticas relacionadas con cada tipo de imágenes para determinar qué herramientas son apropiadas para segmentar cada tipo de imágenes.
- Capítulo 5: analiza los resultados de incorporar información contextual al proceso de umbralización de imágenes mediante la curva de energía. Para ello se evalúa el desempeño de una gran cantidad de algoritmos evolutivos aplicados a la segmentación de la curva de energía haciendo énfasis en el desempeño de técnicas del estado del arte.
- Capítulo 6: propone una estrategia de umbralización sin asistencia usando algoritmos evolutivos multiobjetivo (MOEAs) para determinar simultáneamente la cantidad óptima de umbrales y su mejor ubicación. Además, se presenta la mejora del algoritmo CSA para su aplicación y funcionamiento en problemas multiobjetivo.
- Capítulo 7: discute las conclusiones y detalla futuras líneas de trabajo.

1.7 Resumen

Este capítulo primero se dedica a presentar los antecedentes que motivan y dan origen a los problemas que se busca dar solución en esta investigación. Además, en esta sección se describen los objetivos específicos y generales, así como la metodología que dictará el curso de la investigación. En la última parte del capítulo se describen las aportaciones presentadas a la comunidad científica como fruto de esta investigación.

2 Marco teórico y estado del arte

En este capítulo se presenta una revisión de los conceptos necesarios para abordar las problemáticas planteadas en el capítulo uno dando especial atención a los temas de optimización, procesamiento de imágenes y a las metodologías necesarias para evaluar los aportes realizados en la investigación que se plasma en esta tesis. Pese a que existe un abundante contenido relacionado con los temas de este trabajo, sólo se han seleccionado las contribuciones más destacadas, tanto teóricas como prácticas, que contribuyen a la tesis. Así, en las subsecuentes secciones el lector encontrará la teoría necesaria para comprender los siguientes capítulos.

2.1 Optimización

La palabra optimizar tiene como significado buscar la mejor manera de realizar una actividad. Desde el punto de vista matemático una función $f(x)$ es optimizada cuando se encuentran los argumentos x que logran generar el resultado mínimo o máximo posible para la función dada. La optimización de una función matemática resulta útil cuando dicha función modela un proceso de interés. En este caso, la función puede denominarse función de coste, función objetivo o función de aptitud. Así, los argumentos x reciben el nombre de vector de diseño o solución. De forma genérica la optimización se describe como:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f_i(\mathbf{x}), \quad i = 1, 2, \dots, M \\ \text{sujeto a } \phi_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, J \\ \psi_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K \end{aligned} \tag{2.1}$$

donde $f_i(\mathbf{x})$, $\phi_j(\mathbf{x})$ y $\psi_k(\mathbf{x})$ son funciones del vector de diseño

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T \quad (2.2)$$

Los componentes x_i de \mathbf{x} se conocen como variables de decisión y pueden ser reales, continuas, discretas o una mezcla de ambas. Las funciones $f_i(\mathbf{x})$ donde $i = 1, 2, \dots, M$ se denominan funciones objetivo, en el caso de que $M = 1$ sólo existe un objetivo. En el caso de $M > 1$ el problema recibe la denominación multiobjetivo. Al resultado de la evaluación de un vector de diseño en una función objetivo se le conoce como valor de aptitud. El espacio generado por la combinación de las variables de decisión se denomina espacio de búsqueda \mathbb{R}^n , mientras que el espacio formado por las funciones objetivo se conoce como espacio de soluciones.

Las funciones objetivo pueden ser funciones lineales o no lineales. Las igualdades para ϕ_j y las desigualdades para ψ_k son conocidas como restricciones. Es importante mencionar que las desigualdades se pueden reescribir de la forma ≥ 0 , y también es posible formular las funciones objetivo como un problema de maximización. Esto es posible ya que la maximización de $f(\mathbf{x})$ es equivalente a la minimización de $-f(\mathbf{x})$ y cualquier desigualdad $g(\mathbf{x}) \leq 0$ es equivalente a $-g(\mathbf{x}) \geq 0$. Para las restricciones, el caso más simple para una variable de decisión x_i es $x_{i \min} \leq x_i \leq x_{i \max}$. A estas restricciones se les conoce como límites.

Los optimizadores sólo proporcionan variables de decisión al sistema y observan los resultados. Los optimizadores entonces cambian iterativa y estocásticamente las entradas del sistema basados en la retroalimentación obtenida al momento hasta la satisfacción de un criterio final. El proceso de la modificación de las variables de decisión o soluciones suele estar definido por el mecanismo correspondiente de cada algoritmo.

Dependiendo de las relaciones entre las entradas y salidas, el espacio de soluciones puede ser unimodal o multimodal. En un espacio unimodal, no hay soluciones locales (colinas o valles). En el espacio multimodal, sin embargo, existen múltiples soluciones locales, una de las cuales es el óptimo global. Merece la pena mencionar aquí que los espacios de búsqueda y soluciones de problemas del mundo real podrían estar cambiando. En este caso, se requiere una optimización dinámica para seguir la evolución del óptimo global. Además, algunos espacios de soluciones pueden tener múltiples óptimos globales con valores de aptitud iguales. Un algoritmo de optimización debería ser capaz de descubrir regiones aisladas e identificar los óptimos prometedores. Debe tenerse en cuenta que cada espacio de búsqueda tiene límites, que pueden considerarse como un conjunto de restricciones que limitan los rangos superior e inferior de los parámetros. Un algoritmo de optimización debe estar equipado con mecanismos adecuados para reubicar las soluciones que violan las regiones permitidas del espacio de búsqueda.

Al momento de optimizar la no linealidad y la multimodalidad son las principales problemáticas ya que provocan que los métodos tradicionales, tales como el ascenso de colinas o escalada simple, alcancen óptimos locales. Otro problema surge cuando el número de variables de decisión crece considerablemente creando espacios de soluciones descomunales que tomaría una gran cantidad de recursos computacionales para su exploración exhaustiva.

2.2 Algoritmos metaheurísticos para optimización

En términos generales, la mayoría de los algoritmos metaheurísticos, en especial los algoritmos evolutivos (EAs) tienen marcos de trabajo muy similares. En primer lugar, inician el proceso de optimización creando un conjunto inicial de soluciones aleatorias para un problema determinado. Luego, este conjunto se evalúa de forma iterativa por una o más funciones objetivo asociadas al problema y evolucionan para minimizar o maximizar el o los objetivos. Aunque este marco es muy simple, la optimización de problemas del mundo real requiere considerar y abordar varios temas de los cuales los más importantes son: óptimos locales, coste computacional de la evaluación de funciones, restricciones, múltiples objetivos e incertidumbres.

El estancamiento en un óptimo local es un fenómeno muy común cuando se optimiza, en especial en problemas reales, los cuales tienen espacios de búsqueda en su mayoría desconocidos que pueden contener muchas soluciones subóptimas (Addis, Locatelli, & Schoen, 2005). En este caso, el algoritmo queda atrapado en una de las soluciones locales y asume que es la solución global. Aunque los operadores estocásticos de los EAs mejoran la capacidad local de evasión de óptimos locales en comparación con los enfoques matemáticos deterministas de optimización, el estancamiento en soluciones subóptimas también puede ocurrir en cualquier EA. Los EAs son también, en su mayoría, paradigmas basados en poblaciones. Esto significa que evalúan y mejoran iterativamente un conjunto de soluciones, comúnmente llamado población o enjambre, en lugar de una solución única. Aunque esto también mejora la evasión de óptimos locales, la optimización de problemas costosos con EAs a veces no es viable debido a la necesidad de un gran número de evaluaciones de la función objetivo. En este caso, se deben diseñar mecanismos para disminuir el número requerido de accesos a la función objetivo. Las restricciones son otra dificultad de los problemas reales (Coello, 2002), donde el espacio de búsqueda puede dividirse en dos regiones: factible e inviable. Los agentes de búsqueda de los EAs deben estar equipados con mecanismos adecuados para evitar todas las regiones no factibles y explorar las áreas factibles para encontrar el óptimo global viable. El manejo de las restricciones requiere mecanismos específicos y ha sido un tema popular entre los investigadores.

2.2.1 Algoritmos evolutivos mono-objetivo

De acuerdo con Mirjalili(2019), los algoritmos evolutivos mono-objetivo, *evolutionary algorithm* (EA), estiman el óptimo global para problemas de un solo objetivo. En un problema de objetivo único hay un conjunto de valores óptimos para los parámetros que conduce al valor objetivo óptimo. La función de objetivo es sólo un componente de un problema de optimización. Otros componentes son las entradas (parámetros, variables) y las restricciones. De esta forma, las entradas son las incógnitas desde la perspectiva del optimizador y producen un valor de aptitud (*fitness*) para cada combinación de entradas. La tarea principal de un algoritmo evolutivo es encontrar los valores óptimos de las entradas.

Las restricciones pueden ser consideradas como entradas secundarias para el sistema, pero su impacto es diferente al de las entradas primarias. Las restricciones indican los límites de un sistema y deben considerarse para encontrar soluciones factibles. De hecho, las restricciones definen qué conjunto de valores para las entradas son válidos. El objetivo puede calcularse para cualquier conjunto de variables de entrada. Sin embargo, una solución, y su valor objetivo, no es aceptable en caso de violación de las restricciones. El algoritmo de optimización no tiene como objetivo encontrar valores óptimos para las restricciones, pero debe tenerlos en cuenta para encontrar soluciones viables.

2.2.2 Algoritmos evolutivos multiobjetivo

Los problemas reales de ingeniería a menudo tienen múltiples objetivos. La optimización en un espacio de búsqueda multiobjetivo es muy diferente y requiere consideraciones especiales en comparación con un espacio de búsqueda mono-objetivo. En un problema con un único objetivo, sólo hay que optimizar una función objetivo y encontrar una única solución global. Sin embargo, en los problemas multiobjetivo ya no hay una solución única para el problema, y se debe encontrar un conjunto de soluciones que representen las mejores relaciones o compromisos entre los múltiples objetivos, el conjunto óptimo de Pareto.

Los algoritmos evolutivos multiobjetivo (MOEAs) están diseñados como métodos de optimización que requieren encontrar soluciones óptimas para dos o más objetivos. A diferencia de los EAs, los MOEAs generan como resultados un conjunto de soluciones donde cada solución representa un compromiso entre objetivos. La idea principal subyacente detrás de la optimización multiobjetivo fue introducida por Schafer (Srinivas & Deb, 1995) con la adición de metodologías evolutivas que proporcionan una funcionalidad extra como la evasión de óptimos locales y estrategias de búsqueda sin gradientes. Algunos de los métodos multiobjetivo más relevantes incluyen el Algoritmo Genético de Clasificación No Dominante (NGSA) (Kalyanmoy Deb, 1999), NGSA-II (Kalyanmoy Deb, Member, Pratap, Agarwal, & Meyarivan, 2002), la Optimización de Enjambres de Partículas Multiobjetivo (MOPSO) (Coello et al.,

2004) y el Algoritmo Evolutivo Multiobjetivo basado en la Descomposición (MOEA/D) (Q. Zhang & Li, 2007). Estos métodos pueden optimizar simultáneamente dos o más objetivos a la vez que mantienen la relación entre los objetivos a través del compromiso entre los objetivos existentes.

La diferencia más notable entre MOEAs y EAs es el número de objetivos a ser optimizados; como el nombre sugiere, el primero maneja dos o más objetivos mientras que el segundo sólo uno. No obstante, los MOEAs comparten mecanismos y operadores con las EAs. La mayoría de los MOEAs fueron diseñados como una extensión de un EA exitoso; como resultado, ambos dependen de poblaciones de soluciones para explorar el espacio de búsqueda. En cada iteración, el valor de aptitud o calidad de cada solución está determinado por la función objetivo. En los problemas de un solo objetivo, el valor de aptitud puede compararse directamente con un operador relacional para determinar cuál es mayor o menor, mientras que los problemas multiobjetivo no pueden compararse utilizando el mismo operador. Dada una función de vectores $f(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_o(\mathbf{x})]$ y un espacio de soluciones viables Ω , los MOEAs son diseñados para encontrar un vector $\mathbf{x} \in \Omega$ tal que optimice la función $f(\mathbf{x})$. Para ello, las definiciones descritas por Coello et al. (2004) establecen un marco de comparación que incluye los conceptos de dominancia de Pareto, Optimalidad de Pareto y Frente de Pareto.

Primero, en el caso de minimización, la dominancia de Pareto indica que el vector \mathbf{x} domina al vector \mathbf{x}' (denotado como $\mathbf{x} < \mathbf{x}'$) si $f_i < f_i(\mathbf{x})$ para todas las funciones i en f , y existe por lo menos una i tal que $f_i(\mathbf{x}) < f_i(\mathbf{x}')$.

Un vector de decisión \mathbf{x}^* es denominado Pareto óptimo si no existe un vector $\mathbf{x} \in \Omega$ tal que $\mathbf{x} < \mathbf{x}^*$. Siguiendo esta noción, el conjunto óptimo de Pareto se construye como $P^* = \{\mathbf{x} \in \Omega\}$. Finalmente, el frente Óptimo de Pareto se define como $PF^* = \{f(\mathbf{x}) | \mathbf{x} \in P\}$.

Así, Zhou (Zhou et al., 2011) afirma que el objetivo final de la optimización multiobjetivo es encontrar la aproximación más precisa del Frente Óptimo de Pareto con un alto grado de diversidad.

2.3 Clasificación de algoritmos de evolutivos

Independientemente del tipo de espacio de búsqueda y de la función objetivo, los algoritmos de optimización estocástica pueden dividirse en dos clases principales (Shilane et al., 2008): individuales o poblacionales. En la primera clase, un algoritmo de optimización comienza con una solución candidata para un problema dado. Esta solución es entonces evaluada y mejorada hasta la satisfacción de una condición de parada en el proceso iterativo. En

la última clase, sin embargo, se emplea un conjunto de soluciones candidatas para determinar el óptimo global para los problemas de optimización.

De acuerdo con Mirjalili(2019), la ventaja de los algoritmos individuales es la necesidad de un número mínimo de evaluaciones de funciones. De hecho, el número de evaluaciones de funciones es igual al número de iteraciones. Sin embargo, estas técnicas sufren de un estancamiento local óptimo (convergencia prematura). Es muy probable que una solución única atrape a alguna de las soluciones locales en problemas del mundo real. Por el contrario, las técnicas basadas en poblaciones se benefician de una alta exploración del espacio de búsqueda y de una menor probabilidad de quedar atrapadas localmente. Si una solución está dentro de una solución local, otras soluciones ayudarán a evitarla en las siguientes iteraciones. Como desventaja, tales algoritmos requieren más evaluaciones de la función objetivo y son computacionalmente más costosos.

Hay un gran número de algoritmos en cada clase. Los algoritmos basados en individuos más conocidos son *Tabu search* (Glover, 1989), escalada simple o *hill climbing* (Goldfeld, Quandt, & Trotter, 1966) *Iterated Local Search* (ILS) (Lourenço, Martin, & Stützle, 2003), y *Simulated Annealing* (SA) (van Laarhoven & Aarts, 1987). *Tabu Search* es una técnica de búsqueda local mejorada que utiliza memorias a corto, medio y largo plazo para prohibir y truncar soluciones no prometedoras/repetidas. El algoritmo de escalada simple es también otra técnica de búsqueda local y de base individual que inicia la optimización desde una única solución. Este algoritmo entonces intenta iterativamente mejorar la solución cambiando su posición hasta encontrar un valor óptimo. La búsqueda local iterativa (ILS) es un algoritmo mejorado de escalada de simple para disminuir la probabilidad de ser atrapado en óptimos locales. En este algoritmo, el óptimo obtenido al final de cada ejecución se mantiene y se considera como el punto de partida en la siguiente ejecución. El algoritmo SA usa un comportamiento que regula de forma proporcional a una variable llamada factor de enfriamiento la longitud del salto que se realiza en la búsqueda. Así, cuando la “temperatura” es alta se podrán hacer saltos grandes en el espacio de búsqueda que pueden llevar a encontrar soluciones peores, mientras que al final de la búsqueda la temperatura será tan baja que solo permitirá pequeños movimientos que refinarán la solución encontrada. Esto ayuda a SA a promover la exploración del espacio de búsqueda y evita que quede atrapado en los óptimos locales cuando los busca.

En los últimos años, ha habido muchas mejoras en cada uno de los algoritmos mencionados. El objetivo principal de la mayoría de ellas ha sido paliar el principal inconveniente de estas técnicas: la convergencia prematura. A pesar de varias mejoras con aplicaciones exitosas (Boussaïd, Lepagnot, & Siarry, 2013), la naturaleza de tales técnicas

requiere que muestren una menor exploración y evitación de óptimos locales en comparación con los algoritmos basados en poblaciones. Un conjunto de soluciones es capaz de manejar mejor las dificultades de los problemas del mundo real con un gran número de soluciones locales.

Los algoritmos basados en poblaciones mejoran el conjunto de soluciones iniciales en el transcurso de las iteraciones. Esto se hace en dos fases: exploración (diversificación) y explotación (intensificación) (Črepinšek, Liu, & Mernik, 2013). En la primera fase, un algoritmo cambia abruptamente las soluciones candidatas para asegurar que exploren diferentes regiones del espacio de búsqueda. El objetivo principal en la fase de exploración es descubrir las áreas prometedoras del paisaje de búsqueda. En otras palabras, en la fase de exploración se encuentra una estimación aproximada del óptimo global del problema. La exploración puede hacerse con cambios frecuentes o grandes en las soluciones. Existen diferentes operadores de algoritmos para proporcionar un comportamiento exploratorio. En la optimización por enjambre de partículas *Particle Swarm Optimization* (PSO) (Kennedy & Eberhart, 1995), por ejemplo, el peso de inercia mantiene la tendencia de las partículas hacia sus direcciones anteriores y enfatiza la exploración. En los algoritmos genéticos *Genetic Algorithms* (GA) (Goldberg & Holland, 1988), una alta probabilidad de mutación causa cambios aleatorios en los individuos como el principal mecanismo de exploración.

En la fase de explotación, un algoritmo requiere que las soluciones candidatas busquen las mejores soluciones encontradas hasta ahora. La magnitud de los cambios en las soluciones es mucho menor que la exploración. En otras palabras, la búsqueda se realiza localmente en torno a las soluciones más prometedoras encontradas en la fase de exploración. El mecanismo de explotación es diferente dependiendo del tipo de algoritmo. En PSO, por ejemplo, una baja tasa de inercia provoca una baja exploración y una mayor tendencia hacia las mejores soluciones personales/globales obtenidas. Por lo tanto, las partículas convergen hacia los mejores puntos en lugar de navegar al rededor del espacio de búsqueda. El mecanismo que lleva a la explotación de la GA es el operador de cruce. El proceso de cruce causa ligeros cambios aleatorios en los individuos y en la búsqueda local alrededor de las soluciones candidatas. Lo que dificulta el proceso de diseño y utilización de algoritmos basados en la población es que la exploración y explotación están en conflicto. Un simple comportamiento exploratorio resulta en la mayoría de los casos en encontrar soluciones de mala calidad, ya que el algoritmo nunca tiene la oportunidad de mejorar la precisión de las soluciones. Por el contrario, un mero comportamiento de explotación tiene como resultado la retención en soluciones locales, ya que el algoritmo no cambia las soluciones abruptamente. Un buen equilibrio entre exploración y explotación conduce a encontrar el óptimo global para todo tipo de problemas. Sin embargo, el desafío es que la intensidad de estas dos fases en conflicto es difícil de definir y equilibrar.

Encontrar un buen equilibrio entre exploración y explotación es difícil debido a los diferentes espacios de búsqueda de problemas. Para resolver problemas con algoritmos basados en la población, es común utilizar operadores adaptativos para ajustar la exploración y explotación proporcionalmente al número de iteraciones. Esto significa que el algoritmo pasa suavemente de la exploración a la explotación a medida que aumenta el número de iteraciones. Otra técnica popular es promover la exploración en cualquier etapa de la optimización si no hay mejoras en la mejor solución obtenida hasta ahora (o en todas las soluciones) (Mirjalili, 2019).

Para lograr un buen balance entre exploración y explotación se han propuesto en la literatura una gran cantidad de algoritmos evolutivos que se han vuelto significativamente populares debido a que consideran el problema como una caja negra, no necesitan derivadas, la inspiración de los algoritmos es simple y fácil de entender.

Existe una teoría en el campo de la optimización llamada *No Free Lunch* (NFL) (Wolpert & Macready, 1997) que prueba lógicamente que no existe un algoritmo de optimización para resolver todos los problemas de optimización exitosamente. Esto significa que la solución a un problema particular no es necesariamente el mejor algoritmo en funciones de prueba u otros problemas del mundo real. Para resolver un problema puede ser necesario modificar, mejorar o adaptar un algoritmo. Esto hace que el campo de investigación sea muy activo. Es por ello que una clasificación rigurosa y aceptada de los métodos de optimización como tal no existe e incluso hay un gran debate en la literatura, sin embargo, algunos conceptos asociados a los problemas de optimización pueden ser categorizados de acuerdo a los siguientes criterios (Yang, 2010):

- Número de objetivos. Existen dos categorías bajo esta clasificación: problemas con un solo objetivo ($M = 1$) y problemas multiobjetivo ($M > 1$).
- Numero de restricciones $J + K$. Si no existen restricciones en absoluto ($J = K = 0$) es llamado no-restringido. En cualquier otro caso se denomina restringido.
- Superficie de error. Si sólo existe un valle o pico con un óptimo global, entonces ese problema se dice que es unimodal, por el contrario, si la superficie tiene más de una moda se dice que es multimodal.
- Forma de la función. Si las restricciones ϕ_j y ψ_k son todas lineales el problema se convierte en uno linealmente restringido. Si tanto las restricciones como las funciones objetivo son lineales se convierte en un problema de optimización lineal.
- Tipo de valor de las variables de diseño. Si los valores de las variables de diseño son discretos, se dice que es un problema de optimización discreto o combinatorial. Por el

contrario, si son enteros o reales se transforma en un problema de optimización continuo.

- Nivel de determinación. En el caso de que tanto las variables de diseño y los objetivos o restricciones sean exactos y no exista incertidumbre o ruido en sus valores, se dice que son problemas deterministas. Si existe alguna incertidumbre y ruido en las variables de diseño y funciones objetivo o restricciones la optimización se convierte en un problema de optimización estocástico.

2.4 Procesamiento de imágenes y visión artificial

La visión artificial o visión por computador tiene como objetivo proveer a los sistemas informáticos de “ojos” para generar una representación computable del mundo real. Las imágenes son analizadas por algoritmos que permiten tomar decisiones de acuerdo con lo que existe en dicha representación. El uso de sistemas de procesamiento de imágenes permite realizar tareas como la clasificación de imágenes, seguimiento de objetos, reconstrucción, segmentación y extracción de características entre muchas otras. Sin embargo, el alcance de la investigación reflejada en esta tesis se limita al análisis de técnicas de segmentación.

2.4.1 Procesamiento digital de imágenes

Al conjunto de operaciones y técnicas comúnmente aplicadas a imágenes digitales se conoce como procesamiento de imágenes. Comúnmente el término de visión es usado para referirse a dichas técnicas, sin embargo, existe una sutil diferencia entre el procesamiento de imágenes y la visión por computador (Davies, 2017; Pajares Martinsanz & Cruz García, 2007). Por ejemplo, las técnicas de procesamiento de imágenes no están limitadas al rango visible del espectro electromagnético, mientras que los sistemas de visión se limitan a replicar procesos visuales en imágenes visibles para el ojo humano. Así, las herramientas del procesamiento de imágenes pueden ser aplicadas a cualquier tipo de imagen; por ejemplo, imágenes térmicas, resonancias electromagnéticas, rayos gamma, entre muchas otras. En todo caso, las técnicas de procesamiento de imágenes y visión requieren como base de sus procesos alguna representación del mundo físico en la forma de una imagen digital. Las imágenes digitales son capturadas mediante un proceso de muestreo sobre una señal analógica a partir de dispositivos de captura digitales, siendo los más famosos CCDs y CMOS.

2.4.1.1 Representación de imágenes digitales

El proceso de adquisición de una imagen digital incluye necesariamente los procesos de captura, muestreo, cuantificación y codificación. La forma más común de representación de una imagen es mediante una función bidimensional que cuantifica los niveles de intensidad de la

radiación de una porción del espectro electromagnético en una posición espacial determinada. En general, se expresa como la función $I(u, v)$ donde el valor de intensidad se obtiene por el indexado de las posiciones espaciales u y v . Cada posición espacial es denominada píxel existiendo un máximo de N columnas y M filas. Para facilitar su operación se utiliza un modelo basado en una matriz, de la forma definida en la ecuación (2.3).

$$I(u, v) = \begin{bmatrix} I(1,1) & I(2,1) & \cdots & I(N,1) \\ I(1,2) & I(2,2) & \cdots & I(N,2) \\ \vdots & \vdots & \ddots & \vdots \\ I(1,M) & I(2,M) & \cdots & I(N,M) \end{bmatrix} \quad (2.3)$$

2.4.1.2 Relaciones entre píxeles

Dada la naturaleza de la representación de las imágenes digitales, es posible establecer relaciones entre los elementos de la imagen con el objetivo de extraer información relevante. La primera de estas relaciones y de relevancia para esta tesis, es la vecindad de un píxel. Se define como la relación espacial que existe entre un píxel y los píxeles cercanos. Como los píxeles están organizados en forma de una cuadrícula es posible establecer dos tipos de vecindad; la vecindad de 4-vecinos y 8-vecinos.

La vecindad 4-vecinos se forma a partir de los cuatro píxeles adyacentes en las direcciones arriba, abajo, izquierda y derecha (Fig. 2.1, V1, V2, V3 y V4). En cambio, la vecindad 8-vecinos está compuesta por los cuatro píxeles de la vecindad 4-vecinos y los cuatro píxeles más cercanos en las direcciones diagonales (Fig. 2.1, V5, V6, V7 y V8).

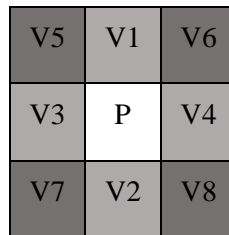


Figura 2.1. Vecindad del píxel P.

2.4.2 Histograma

Los histogramas son herramientas estadísticas que nos permiten visualizar las distribuciones descritas por la frecuencia con la que ocurren un conjunto de eventos. Desde la perspectiva del procesamiento de imágenes, los histogramas consideran como eventos todos los niveles de intensidad posibles en la imagen y a cada uno de ellos se le asocia con la cantidad de apariciones de ese valor de intensidad a lo largo de la imagen. En el caso de las imágenes en

escala de grises, una imagen $I(u, v)$ que contiene valores de intensidad en el intervalo $[0, L-1]$ generará un histograma H con exactamente L diferentes valores. Dado que en las imágenes en escala de grises es común codificar las intensidades mediante 8 bits, se obtiene un total de $L = 2^8 = 256$ niveles de intensidad. Por lo tanto, la frecuencia de ocurrencia del nivel de intensidad i se define como $h(i)$ y es igual al número de píxeles con la intensidad i existentes en la imagen I para todos los valores $0 \leq i < L$. Formalmente se define como:

$$h(i) = \text{card} \{ (u, v) \mid I(u, v) = i \} \quad (2.4)$$

donde $\text{card} \{ \dots \}$ representa el número de elementos. Así, $h(0)$ es el número de píxeles que tienen valor de intensidad 0. De forma similar $h(1)$ expresa el número de píxeles con valor de 1 y de esta forma se continúa hasta $h(255)$. El histograma resultante es presentado en forma de un vector unidimensional de longitud L . En la figura 2.2 podemos observar una imagen junto a su histograma.

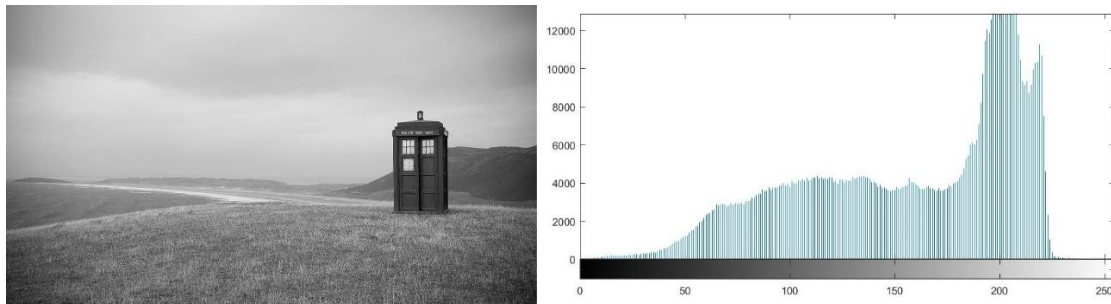


Figura 2.2. Imagen y su histograma correspondiente.

El histograma es ampliamente utilizado para analizar características importantes para evaluar la calidad de una imagen, como el contraste y rango dinámico. Cuando una imagen es capturada bajo malas condiciones de iluminación el histograma mostrará dichos problemas, los cuales pueden ser corregidos mediante procesamiento posteriores.

Los histogramas, a pesar de su gran utilidad, tienen algunas limitantes. Por ejemplo, con el histograma podemos saber cuántos píxeles hay de cierta intensidad, pero no podemos conocer la ubicación espacial de dichos píxeles sólo con los histogramas.

Segmentación de imágenes

Formalmente, el problema de segmentación se establece como un proceso que separa una imagen en regiones representativas de acuerdo con por lo menos un criterio que se comparta dentro de cada región (Gonzalez & Woods, 1992):

Sea $I(u, v)$ una imagen y R la representación completa de la imagen. De acuerdo con la segmentación, es posible decir que R contiene n subregiones R_i , $i = 1, 2, \dots, n$ de tal forma que se cumpla lo siguiente

- a) La segmentación debe ser completa, de forma que cada píxel de la imagen debe estar presente en alguna subregión, es decir $\bigcup_{i=1}^n R_i = R$.
- b) R_i tiene algún tipo de conectividad o característica común.
- c) La segmentación debe ser disjunta, $R_i \cap R_j = \emptyset$ para todo i, j donde $i \neq j$.
- d) $P(R_i) = \text{Verdadero}$ donde $P(\)$ es un predicado lógico sobre los puntos del conjunto R_i y el conjunto vacío \emptyset .
- e) $P(R_i \cup R_j) = \text{Falso}$ para $i \neq j$.

En la mayoría de los sistemas de visión se requiere realizar segmentación. La correcta segmentación puede aportar una mejora en el desempeño de todo el sistema en general ya que es común que otros procesos dependan de los resultados de dicha segmentación. Todos los sistemas de segmentación pueden agruparse en dos categorías: basada en homogeneidad de regiones y basada en discontinuidad entre regiones adyacentes. El hecho de dividir las imágenes en regiones (partición) permite analizar y procesar solo las regiones que sean de interés. Además, la segmentación puede ser usada para reducir el uso de memoria e incrementar la velocidad del sistema ya que se reduce la información redundante.

El objetivo principal de la segmentación es la partición de la imagen en clases homogéneas, donde los elementos de cada clase comparten propiedades comunes como la intensidad o la textura. Actualmente, se han propuesto una gran cantidad de contribuciones en la literatura, que se pueden encontrar en las siguientes investigaciones (Osuna-Enciso et al., 2013; Zaitoun & Aqel, 2015; Zhang, 1996). Una de las técnicas más simples de segmentación es la umbralización de imágenes (UI). La UI utiliza el histograma de escala de grises para seleccionar valores de umbrales que están destinados a separar clases en el histograma de la imagen. La umbralización de dos niveles es el caso más sencillo y sólo utiliza un valor umbral para crear dos clases. Bajo este enfoque, se puede extraer un objeto de su fondo. Mientras que la umbralización de dos niveles es fácil de implementar, la umbralización multinivel (UM) presenta más complicaciones, ya que su objetivo es encontrar varias clases. Los métodos de

segmentación basados en la umbralización pueden dividirse en paramétricos y no paramétricos (Akay & Bahriye, 2013; Hammouche et al., 2010; Liao et al., 2001). Los enfoques paramétricos estiman los parámetros de una función de densidad de probabilidad para describir cada clase, pero este enfoque es computacionalmente costoso. Por el contrario, los enfoques no paramétricos utilizan criterios como la varianza entre clases, la entropía y la tasa de error para encontrar los mejores umbrales (Kapur, Sahoo, & Wong, 1985; Kittler & Illingworth, 1986; Otsu, 1979). Estos criterios son optimizados para encontrar los valores de umbral óptimos, proporcionando métodos robustos y precisos (Sankur, 2004). Otsu propuso un método popular que puede encontrar umbrales maximizando la varianza entre las clases de los niveles de intensidad del primer plano y del fondo (Otsu, 1979). El método de Otsu es considerado uno de los mejores métodos de selección de umbral para imágenes del mundo real (Sahoo et al., 1988). Sin embargo, la formulación de la varianza entre clases es ineficaz en el caso de los umbrales multinivel. A medida que el número de niveles crece, el tiempo computacional se incrementa exponencialmente, y su precisión disminuye con cada nuevo punto de umbral (Sathya & Kayalvizhi, 2011). Otro método ampliamente utilizado ha sido propuesto por Kapur (1985), y maximiza la entropía para medir la homogeneidad de cada clase. Ambos enfoques han sido evaluados en varios contextos, demostrando su eficiencia y precisión (Sathya & Kayalvizhi, 2011).

Además, se ha adoptado el desarrollo de la teoría de la información para explorar el uso de varias entropías con fines de segmentación. Bajo estos enfoques, el objetivo es encontrar criterios de entropía eficientes que con su optimización permitan separar regiones en una imagen. Algunos ejemplos incluyen la entropía de Kapur, la entropía de Tsallis (Portes de Albuquerque, Esquef, & Gesualdi Mello, 2004), y la entropía cruzada (Li & Lee, 1993) por enumerar algunos. Una de las medidas de entropía más importantes es el criterio de la entropía cruzada mínima (MCE) (Li & Lee, 1993), la cual ha sido utilizada ampliamente en la literatura para segmentar imágenes. Este criterio fue introducido por Li y Lee en un algoritmo de segmentación que identifica umbrales minimizando la entropía cruzada entre las regiones de la imagen segmentada. Por otro lado, Yin propuso una técnica de programación recursiva que reduce la magnitud de la computación de la función objetivo del MCE (Yin, 2007).

Como alternativa a las técnicas paramétricas, el problema de la UM también se ha tratado mediante algoritmos evolutivos. Tales acercamientos produjeron varias aplicaciones de la UM seleccionando diversas técnicas de computación evolutivas y optimizando diversos criterios, tales como la optimización con el enjambre de partículas (PSO) y el algoritmo de la luciérnaga (FF) para usar como criterio la entropía cruzada (Horng & Liou, 2011). Otro ejemplo es el uso de la colonia artificial de abejas (ABC) y PSO con el criterio de Kapur (Akay

& Bahriye, 2013) y por último el uso del algoritmo de forrajeo bacteriano modificado (MBFO) con la entropía de Tsallis (Tang, Xiao, Wu, Yang, & Luo, 2017).

2.4.3 Umbralización

Entre las técnicas de segmentación existentes, la umbralización destaca por su implementación simple y la calidad de sus resultados. La idea general es elegir los valores de intensidad, esto es los denominados umbrales que separen óptimamente el histograma de una imagen. En la figura 2.3 podemos observar cómo influye la selección del valor de un umbral en el resultado de la segmentación. Para hacer más evidente la contribución del valor de umbral se muestra un ejemplo de umbralización bi-nivel.

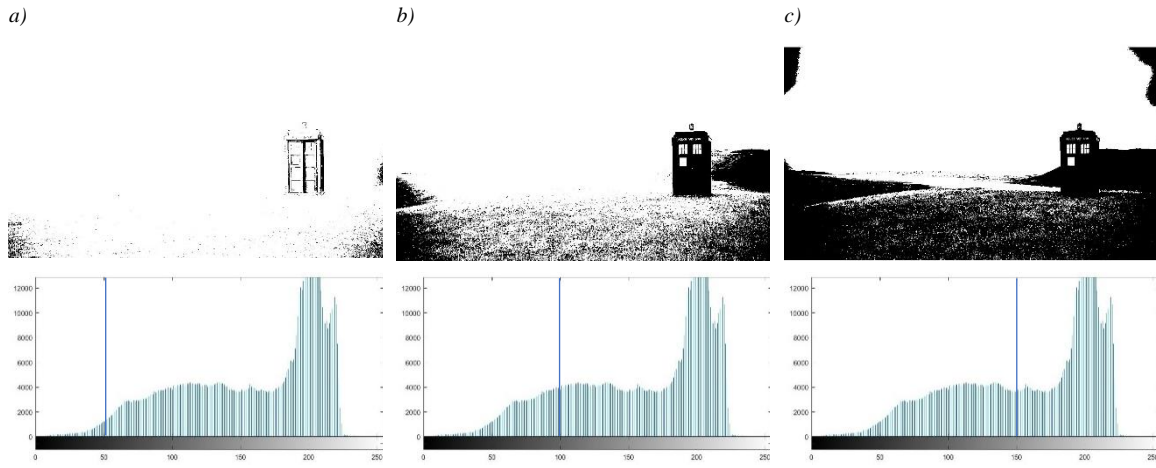


Figura 2.3. Comparación del efecto de la selección del umbral. a) umbral en 50, b) umbral en 100 y c) umbral en 150.

Las técnicas de umbralización funcionan integrando un criterio que evalúa la calidad de los umbrales y un algoritmo que propone tales valores. De forma general, un algoritmo evolutivo puede utilizarse para proponer valores de umbral y utilizar un criterio no paramétrico para determinar la calidad de tal propuesta. El algoritmo evolutivo iterará hasta que se encuentre un conjunto de umbrales cercanos al óptimo. Una vez que el algoritmo evolutivo encuentra los mejores valores de umbrales, es posible proceder a realizar la umbralización. Para ello, en esta investigación se utiliza la siguiente regla para dos umbrales y tres clases:

$$I_s(u, v) = \begin{cases} I_{Gr}(u, v) & \text{si } I_{Gr}(u, v) \leq th_1 \\ th_1 & \text{si } th_1 < I_{Gr}(u, v) \leq th_2 \\ I_{Gr}(u, c) & \text{si } I_{Gr}(u, v) > th_2 \end{cases} \quad (2.5)$$

donde $I_s(u, v)$ es el valor del píxel de la imagen segmentada, $I_{Gr}(u, v)$ es el valor del píxel de la imagen original en la posición u, v . th_1 y th_2 son los valores de umbral obtenidos por el algoritmo evolutivo. La Ec. 2.5 puede ser fácilmente ampliada para considerar más umbrales y

clases quedando de forma genérica como la 2.6. En la figura 2.4 se muestra el resultado de segmentar una imagen con cuatro clases generadas por tres umbrales.

$$I_s(u,v) = \begin{cases} I_{Gr}(u,v) & \text{si } I_{Gr}(u,v) \leq th_1 \\ th_{i-1} & \text{si } th_{i-1} < I_{Gr}(u,v) \leq th_i, \quad i = 2, 3, \dots, nt-1 \\ I_{Gr}(u,v) & \text{si } I_{Gr}(u,v) > th_{nt} \end{cases} \quad (2.6)$$

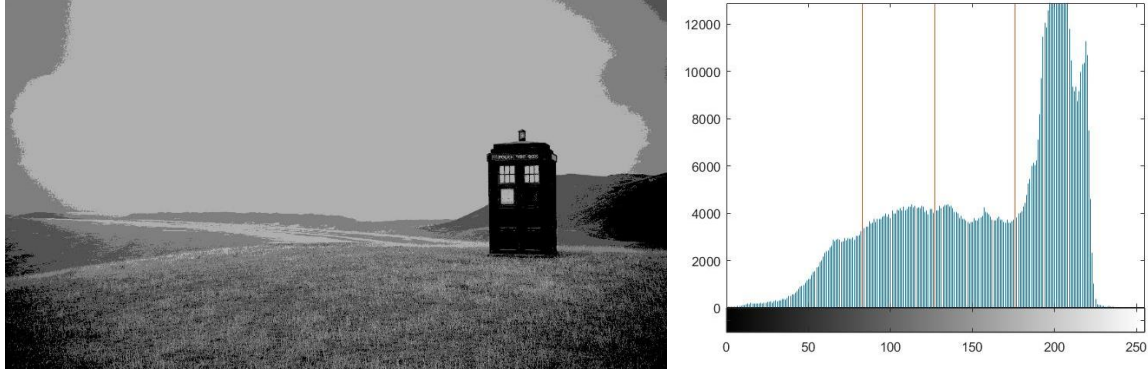


Figura 2.4. Ejemplo de umbralización multinivel con cuatro clases y tres umbrales ($th = [83 \ 127 \ 176]$).

En las siguientes subsecciones se detallará la formulación de los tres criterios de umbralización de interés para la investigación plasmada en esta tesis iniciando con la varianza entre clases, continuando con la entropía máxima y por último la entropía cruzada mínima.

2.4.4 Varianza entre clases

La técnica más popular para la umbralización fue propuesta por Otsu (1979). Esta técnica segmenta la imagen mediante la maximización de la diferencia entre las varianzas de las distintas clases. El valor de intensidad de cada píxel de una imagen tiene asociada una distribución de probabilidad de acuerdo con la siguiente ecuación:

$$Ph_i = \frac{h_i}{NP}, \quad \sum_{i=1}^{NP} Ph_i = 1 \quad (2.7)$$

donde i representa la intensidad de un píxel tal que $(0 \leq i \leq L-1)$, NP es el número total de píxeles presentes en la imagen, h es el histograma de la imagen h_i indica la cantidad de apariciones del valor de intensidad i . El histograma está asociado con una distribución de probabilidad Ph_i . En el caso más simple, la segmentación bi-nivel se realiza mediante:

$$C_1 = \frac{Ph_1}{\omega_0(th)}, \dots, \frac{Ph_{th}}{\omega_0(th)} \quad \text{y} \quad C_2 = \frac{Ph_{th+1}}{\omega_1(th)}, \dots, \frac{Ph_L}{\omega_1(th)} \quad (2.8)$$

donde $\omega_0(th)$ y $\omega_1(th)$ son las distribuciones de probabilidad para C_1 y C_2

$$\omega_0(th) = \sum_{i=1}^{th} Ph_i, \quad \omega_1(th) = \sum_{i=th+1}^L Ph_i \quad (2.9)$$

Así, las dos clases generadas estarán definidas por μ_0 y μ_1 calculadas como:

$$\mu_0 = \sum_{i=1}^{th} \frac{iPh_i}{\omega_0(th)}, \quad \mu_1 = \sum_{i=th+1}^L \frac{iPh_i}{\omega_1(th)} \quad (2.10)$$

Después, las varianzas σ_1 y σ_2 de C_1 y C_2 respectivamente, son calculadas mediante:

$$\sigma_1 = \omega_0 (\mu_0 + \mu_T)^2, \quad \sigma_2 = \omega_1 (\mu_1 + \mu_T)^2 \quad (2.11)$$

donde $\mu_T = \omega_0 \mu_0 + \omega_1 \mu_1$ y $\omega_0 + \omega_1 = 1$. Finalmente, el operador de la varianza de Otsu σ^2 puede calcularse usando la Ec. 2.12. Se hace notar que el número dos es parte del operador y no indica exponenciación.

$$\sigma^2 = \sigma_1 + \sigma_2 \quad (2.12)$$

Desde el punto de vista de la optimización, la función de aptitud $f_{Otsu}(th)$ se genera para maximizar la varianza de Otsu, donde $\sigma^2(th)$ indica el operador de Otsu para el valor de umbral th :

$$f_{Otsu}(th) = \max(\sigma^2(th)), \quad 0 \leq th \leq L-1 \quad (2.13)$$

Para el enfoque multinivel se necesitan nt umbrales para dividir la imagen original en $nt+1$ clases. Este escenario involucra el cálculo de nt varianzas y otros elementos. Así, la función de aptitud $f_{Otsu}(th)$ es reescrita considerando más de un umbral:

$$f_{Otsu}(\mathbf{th}) = \max(\sigma^2(\mathbf{th})), \quad 0 \leq th_i \leq L-1, \quad i = 1, 2, \dots, nt \quad (2.14)$$

donde $\mathbf{th} = [th_1, th_2, \dots, th_{nt-1}]$ es un vector que contiene los diferentes valores de umbral a ser aplicados. La varianza es también actualizada para incorporar más umbrales de la forma:

$$\sigma^2 = \sum_{j=1}^{nt} \sigma_j = \sum_{j=1}^{nt} \omega_j (\mu_j - \mu_T)^2 \quad (2.15)$$

donde j es el índice de cada una de las clases a ser segmentadas, ω_i^c y μ_j^c son la probabilidad de ocurrencia y la media de la clase respectivamente. Los valores ω_i pueden determinarse para la umbralización multinivel como se muestra a continuación:

$$\begin{aligned}
\omega_0(th) &= \sum_{i=1}^{th_1} Ph_i \\
\omega_1(th) &= \sum_{i=th_1+1}^{th_2} Ph_i \\
&\vdots \\
\omega_{k-1}(th) &= \sum_{i=th_{k-1}+1}^L Ph_i
\end{aligned} \tag{2.15}$$

Además, las ecuaciones de la media se actualizan a:

$$\begin{aligned}
\mu_0 &= \sum_{i=1}^{th_1} \frac{iPh_i}{\omega_0(th_1)} \\
\mu_1 &= \sum_{i=th_1+1}^{th_2} \frac{iPh_i}{\omega_1(th_2)} \\
&\vdots \\
\mu_{k-1} &= \sum_{i=th_{k-1}+1}^{nt} \frac{iPh_i}{\omega_{k-1}(th_k)}
\end{aligned} \tag{2.16}$$

2.4.5 Entropía máxima

El método propuesto por Kapur para encontrar valores de umbral óptimos usa la distribución de probabilidad del histograma de la imagen en combinación con el concepto de entropía. La idea principal es encontrar la mejor combinación de umbrales que maximiza la entropía (Kapur et al., 1985). En este caso, cuando el umbral óptimo, th , separa el histograma en dos clases la entropía es máxima. Siguiendo su propuesta inicial como segmentación bi-nivel la función de aptitud se define como:

$$f_{Kapur}(th) = H_1 + H_2 \tag{2.17}$$

donde las entropías H_1 y H_2 se calculan como:

$$H_1 = \sum_{i=1}^{th} \frac{Ph_i}{\omega_0} \ln \left(\frac{Ph_i}{\omega_0} \right), \quad H_2 = \sum_{i=th+1}^L \frac{Ph_i}{\omega_1} \ln \left(\frac{Ph_i}{\omega_1} \right) \tag{2.18}$$

La distribución de probabilidad Ph_i del nivel i se determina como:

$$Ph_i = \frac{h_i}{NP}, \quad \sum_{i=1}^{NP} Ph_i = 1 \tag{2.19}$$

Además, la distribución de probabilidad de cada clase se define como la suma de la probabilidad de cada uno de sus componentes:

$$\omega_0(th) = \sum_{i=1}^{th} Ph_i \quad (2.20)$$

Se busca que un algoritmo evolutivo encuentre el mejor umbral th^* con la maximización de la entropía de Kapur:

$$th^* = \arg \max_{th} f_{Kapur}(th) \quad (2.21)$$

De forma similar al método de Otsu, el método de Kapur puede extenderse para incorporar múltiples umbrales. En tal caso, la imagen se divide en $nt+1$ clases. Bajo tales condiciones la función de aptitud se redefine como:

$$f_{Kapur}(\mathbf{th}) = \sum_{i=1}^{nt} H_i \quad (2.22)$$

donde $\mathbf{th} = [th_1, th_2, \dots, th_{nt}]$ es el vector que contiene los valores de umbral. Cada valor de entropía se calcula de forma independiente de acuerdo con su respectivo umbral. Así, la Ec. 2.22 se define para nt entropías:

$$\begin{aligned} H_1 &= \sum_{i=1}^{th_1} \frac{Ph_i}{\omega_0} \ln \left(\frac{Ph_i}{\omega_0} \right) \\ H_2 &= \sum_{i=th_1+1}^{th_2} \frac{Ph_i}{\omega_1} \ln \left(\frac{Ph_i}{\omega_1} \right) \\ &\vdots \\ H_k &= \sum_{i=th_{k-1}+1}^L \frac{Ph_i}{\omega_{k-1}} \ln \left(\frac{Ph_i}{\omega_{k-1}} \right) \end{aligned} \quad (2.23)$$

2.4.6 Entropía cruzada mínima

La entropía cruzada mínima, también conocida como divergencia, es una métrica teórica de información usada para medir la cantidad de información de un proceso aleatorio (Tang, Yuan, Sun, Yang, & Gao, 2011). En otras palabras, ayuda a medir la distancia entre dos distribuciones probabilísticas (Pal, 1996). La entropía cruzada fue introducida por Kullback (1968). En su forma general se define como:

$$D(\mathbf{B}, \mathbf{C}) = \sum_{k=1}^N b_k \log \left(\frac{b_k}{c_k} \right) \quad (2.24)$$

donde $\mathbf{B} = \{b_1, b_2, \dots, b_N\}$ y $\mathbf{C} = \{c_1, c_2, \dots, c_N\}$ son dos distribuciones probabilísticas del mismo conjunto. D es la entropía cruzada. La entropía cruzada puede ser interpretada como la esperanza de cambio en el contenido de la información cuando es usado B en lugar de C (Li &

Lee, 1993). Es importante mencionar que un valor mayor de entropía cruzada representa mayor incertidumbre en el proceso aleatorio.

En el procesamiento de imágenes, y específicamente en la segmentación de imágenes, es posible seleccionar un conjunto de valores de umbral $\mathbf{th} = [th_1, th_2, \dots, th_m]$ a partir del histograma h de la imagen. Los valores contenidos en \mathbf{th} deben ser tales que logran minimizar la entropía cruzada que existe entre las distintas clases de la imagen segmentada. El ejemplo más simple de la segmentación de imágenes es usando un único umbral $\mathbf{th} = [th_1]$ el cual, una vez seleccionado permite separar los píxeles de la imagen de acuerdo con la siguiente regla:

$$\mathbf{I}_s(i, j) = \begin{cases} \mu(1, th_1) & \text{si } \mathbf{I}_{or}(i, j) < th_1, \\ \mu(th_1, L-1) & \text{si } \mathbf{I}_{or}(i, j) \geq th_1 \end{cases} \quad (2.25)$$

donde L es el máximo valor de intensidad del histograma de la imagen ($L=256$ en una imagen en escala de grises con 8 bits de representación). La regla presentada en la Ec. 2.25 puede ser fácilmente expandida para trabajar con más de un umbral \mathbf{th} . Considerando el histograma h de la imagen original, el valor normalizado μ para un rango específico restringido por a y b se calcula como:

$$\mu(a, b) = \frac{\sum_{i=a}^{b-1} ih(i)}{\sum_{i=a}^{b-1} h(i)}, \quad i = 0, 1, \dots, L-1 \quad (2.26)$$

Para verificar si la imagen original fue correctamente segmentada es necesario aplicar la ECM para imágenes digitales. La entropía cruzada mínima se determina usando el método propuesto por Li y Lee (1993) estando definida para un solo umbral ($\mathbf{th} = [th_1]$) de forma:

$$D(\mathbf{th}) = \sum_{i=1}^{th_1-1} ih(i) \log \left(\frac{i}{\mu(1, th_1)} \right) + \sum_{i=th_1}^L ih(i) \log \left(\frac{i}{\mu(th_1, L+1)} \right) \quad (2.27)$$

Así, el objetivo es encontrar el mejor conjunto de valores de umbral que minimicen la entropía cruzada. Para lograr este fin, la EC. 2.28 se define como función objetivo.

$$\mathbf{th}_{opt} = \arg \min_{\mathbf{th}} (D(\mathbf{th})) \quad (2.28)$$

La complejidad computacional para encontrar un umbral óptimo es $O(nt \cdot L^2)$ pero se vuelve computacionalmente costosa para realizar este proceso con múltiples umbrales. Para nt umbrales la complejidad crece a $O(nt \cdot L^{nt+1})$ (Tang et al., 2011).

Como se menciona anteriormente, la ECM es computacionalmente costosa para la segmentación multinivel. Para reducir el esfuerzo computacional, se ha propuesto una versión que hace uso de la programación recursiva para obtener resultados más rápidos en la umbralización de imágenes (Hammouche, Diaf, & Siarry, 2008; Yin, 2007). En la versión recursiva de la ECM la Ec. (2.27) puede ser reescrita como:

$$D(\mathbf{th}) = \underbrace{\sum_{i=1}^L ih(i) \log(i)}_{\text{Entropía de la imagen}} - \underbrace{\sum_{i=1}^{th_1-1} ih(i) \log(\mu(1, th_1))}_{\text{Entropía de 1 a } th_1-1} - \underbrace{\sum_{i=th_1}^L ih(i) \log(\mu(th_1, L+1))}_{\text{Entropía de } th_1 \text{ a } L+1} \quad (2.29)$$

La Eq. (2.27) se define para un solo umbral th y dos clases. El primer término es constante para una imagen dada y el resto de los elementos dependen directamente en el umbral seleccionado. Considerando tales hechos, la función objetivo puede ser reescrita como:

$$\begin{aligned} \varphi(\mathbf{th}) &= - \sum_{i=1}^{th_1-1} ih(i) \log(\mu(1, th_1)) - \sum_{i=th_1}^L ih(i) \log(\mu(th_1, L+1)) \\ &= - \left(\sum_{i=1}^{th_1-1} ih(i) \right) \log \left(\frac{\sum_{i=1}^{th_1-1} ih(i)}{\sum_{i=1}^{th_1-1} h(i)} \right) - \left(\sum_{i=th_1}^L ih(i) \right) \log \left(\frac{\sum_{i=th_1}^L ih(i)}{\sum_{i=th_1}^L h(i)} \right) \\ &= -m^1(1, th_1) \log \left(\frac{m^1(1, th_1)}{m^0(1, th_1)} \right) - m^1(th_1, L+1) \log \left(\frac{m^1(th_1, L+1)}{m^0(th_1, L+1)} \right) \end{aligned} \quad (2.30)$$

aquí se calculan los valores del punto del momento cero $m^0(a, b) = \sum_{i=a}^{b-1} h(i)$ y los valores de punto del primer momento $m^1(a, b) = \sum_{i=a}^{b-1} ih(i)$ para un rango parcial de valores del histograma.

La Ec. 2.30 puede extenderse para la segmentación multinivel considerando un conjunto de umbrales denotados como \mathbf{th} , donde nt es el número de umbrales a ser encontrados. Para mayor conveniencia se han agregado dos umbrales inoperativos siendo $th_0 \equiv 0$ y $th_{nt+1} \equiv L+1$. Es muy importante mantener el orden de tales valores de forma que: $th_0 < th_1 < \dots < th_{nt} < th_{nt+1}$. La programación recursiva se aplica en la formulación de la función objetivo para la segmentación multinivel usando ECM de forma:

$$\varphi(\mathbf{th}) = m^1(th_{i-1}, th_i) \log \left(\frac{m^1(th_{i-1}, th_i)}{m^0(th_{i-1}, th_i)} \right), \quad \mathbf{th} = [th_1, th_2, \dots, th_{nt}], \quad i = 1, 2, \dots, nt \quad (2.31)$$

Con el uso de la Ec. 2.31 la complejidad es reducida de $O(nt \cdot L^{nt+1})$ a $O(nt \cdot L^{nt})$ (Tang et al., 2011).

2.5 Criterios de comparación

La calidad de una imagen segmentada puede compararse visualmente. Sin embargo, en muchos casos las diferencias entre dos imágenes segmentadas pueden escapar a una inspección visual. Además, es necesario cuantificar la calidad de una imagen segmentada para poder establecer un marco de comparación apropiado.

En la literatura se han presentado distintos criterios para comparar los resultados obtenidos sobre una imagen; entre ellos podemos encontrar los que analizan la calidad de una imagen con herramientas de procesamiento de señales, dado que una imagen puede ser interpretada como una señal bidimensional. Las dos métricas que destacan son el error cuadrático medio (MSE) y la relación señal a ruido de pico (PSNR) (Horng & Liou, 2011). El objetivo fundamental de ambas medidas es comparar la imagen original frente a la imagen segmentada para cuantificar el error o diferencia entre ambas.

2.5.1 MSE

En el error cuadrático medio la diferencia entre los valores de cada píxel de la imagen original I_{Gr} y la segmentada I_s se considera como:

$$MSE = \frac{1}{ro \times co} \sum_{u=1}^{ro} \sum_{v=1}^{co} (I_{Gr}(u,v) - I_s(u,v))^2 \quad (2.32)$$

donde ro y co son los números de filas y columnas respectivamente.

2.5.2 PSNR

A pesar de que el MSE es capaz de cuantificar la disimilitud entre dos imágenes, la relación de señal a pico (PSNR) considera los niveles máximos de intensidad posibles en un píxel para así generar un valor en decibelios (dB). Además, en su definición está considerado el error cuadrático medio, por lo que el uso de ambas resulta redundante:

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right), (dB) \quad (2.33)$$

2.5.3 SSIM

Sin embargo, tanto el MSE como el PSNR consideran sólo las relaciones numéricas entre las intensidades presentes en la imagen. En la literatura se han propuesto métricas para analizar aspectos desde un punto de vista más cercano a la visión humana. Para ello se han presentado medidas que no sólo consideran los valores de los píxeles, también analizan cómo se han preservado elementos que dan significado a la imagen. Por ejemplo, el índice de similitud de estructuras (SSIM) se utiliza para determinar cómo han sido afectadas las estructuras de una imagen después de ser alterada (Wang, Bovik, Sheikh, & Simoncelli, 2004). SSIM toma valores

entre 0 y 1, siendo este último número el que indica una exacta similitud. Para su cómputo se usa la siguiente expresión:

$$SSIM(I_{or}, I_{th}) = \frac{(2\mu_{I_{or}}\mu_{I_{th}} + C1)(2\sigma_{I_{or}I_{th}} + C2)}{(\mu_{I_{or}}^2 + \mu_{I_{th}}^2 + C1)(\sigma_{I_{or}}^2 + \sigma_{I_{th}}^2 + C2)} \quad (2.34)$$

$$\sigma_{I_{th}I_{or}} = \frac{1}{N-1} \sum_{i=1}^N (I_{or_i} + \mu_{I_{or}})(I_{th_i} + \mu_{I_{th}})$$

En la Ec. 2.34 la media de la imagen original es $\mu_{I_{or}}$ mientras que la media de la imagen segmentada está representada por $\mu_{I_{th}}$. De la misma forma, para cada imagen, los valores $\sigma_{I_{Gr}}$ y $\sigma_{I_{th}}$ corresponden a sus desviaciones estándar. $C1$ y $C2$ son constantes usadas para evitar inestabilidad cuando $\mu_{I_{Gr}}^2 + \mu_{I_{th}}^2 \approx 0$. En esta investigación, los valores de $C1$ y $C2$ se establecen como 0.065 de acuerdo a Agrawal et al. (2013).

2.5.4 FSIM

Otro método que se ha utilizado para determinar la calidad de una imagen segmentada es el índice de similitud de características (FSIM) (Zhang, 2011) y tiene como objetivo analizar la calidad de las características en la imagen después de la segmentación. La expresión que define el FSIM es:

$$FSIM = \frac{\sum_{w \in \Omega} S_L(w) PC_m(w)}{\sum_{w \in \Omega} PC_m(w)} \quad (2.35)$$

donde Ω representa el dominio completo de la imagen:

$$S_L(w) = S_{PC}(w) S_G(w) \quad (2.36)$$

$$S_{PC}(w) = \frac{2PC_1(w)PC_2(w) + T_1}{PC_1^2(w) + PC_2^2(w) + T_1} \quad (2.37)$$

$$S_G(w) = \frac{2G_1(w)G_2(w) + T_2}{G_1^2(w) + G_2^2(w) + T_2} \quad (2.38)$$

G es la magnitud del gradiente de una imagen y se define como:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.39)$$

siendo G_x y G_y las primeras derivadas direccionales espaciales en x e y .

Mientras que la congruencia de fase PC se determina mediante:

$$PC(w) = \frac{E(w)}{(\varepsilon + \sum_n A_n(w))} \quad (2.40)$$

La magnitud del vector respuesta en w es $E(w)$ y $A_n(w)$ es la amplitud local en la escala n , ε es un entero positivo y finalmente $PC_m(w) = \max(PC_1(w), PC_2(w))$.

2.5.5 Prueba de significancia

Dada la naturaleza estocástica de los algoritmos evolutivos, es necesario utilizar herramientas estadísticas que permitan analizar su comportamiento. Específicamente, los resultados generados por algoritmos estocásticos varían en cada ejecución. Es por ello por lo que su desempeño debe ser analizado en un contexto estadístico. Todos los experimentos realizados en la investigación desarrollada en esta tesis han sido evaluados al menos en 30 evaluaciones independientes para reportar su desempeño promedio.

Además, en muchos casos las soluciones obtenidas por un algoritmo evolutivo A pueden ser muy similares a los resultados del algoritmo B. Para determinar si los resultados de A son significativamente distintos de los de B es necesario aplicar una prueba de significancia estadística. Dado que no es posible garantizar que las soluciones obtenidas por un algoritmo evolutivo siguen una distribución normal, se ha adoptado una prueba de significancia no paramétrica conocida como suma de rangos o prueba de Wilcoxon (García, Molina, Lozano, & Herrera, 2009) la cual se realiza con 35 muestras independientes. Tal prueba permite evaluar las diferencias entre resultados de dos métodos relacionados. El análisis se realiza considerando un nivel de significancia de 5% sobre el mejor valor analizado. Los resultados son evaluados en grupos de pares donde se evalúa uno a uno el algoritmo propuesto contra los que se compara (A vs. B, A vs. C en el caso de que se proponga A). Como hipótesis nula se asume que no existe diferencia entre los valores de los dos algoritmos evaluados. La hipótesis alternativa considera la existencia de una diferencia significativa entre los valores de los dos algoritmos evaluados. Todos los valores p reportados en tablas donde los valores menores que 0.05 (nivel de significancia 5%) indican una evidencia fuerte en contra de la hipótesis nula, indicando que los valores encontrados por el algoritmo evaluado (A) son diferentes a los de los otros algoritmos (B, C) y además no son resultados aleatorios.

2.6 Resumen

En este capítulo se describe la teoría relativa a los dos conceptos más importantes de la tesis: optimización y segmentación. En cada caso se parte de la teoría general hasta llegar a lo específico, prestando especial atención a los algoritmos evolutivos y a la umbralización de imágenes. Por último, se presentan las métricas utilizadas para comparar los resultados de los distintos algoritmos evaluados, así como la descripción de la prueba de significancia usada.

3 Umbralización usando entropía cruzada mínima

El objetivo de este capítulo es mejorar la calidad de las imágenes segmentadas mediante el uso de la entropía cruzada mínima como función objetivo. En este capítulo se propone un nuevo algoritmo de UM para imágenes digitales basado en el algoritmo EMO con el fin de minimizar la entropía cruzada. Por tanto, el capítulo se organiza de forma que en la sección 3.1 se describe EMO, que da paso al proceso de segmentación descrito en la sección 3.2, para finalizar con los resultados obtenidos, sección 3.3 y su análisis en la sección 3.4. En la sección 3.5 se realiza un resumen de lo expuesto en el capítulo. Es posible encontrar más información en el artículo los artículos (Oliva, Hinojosa, et al., 2017; Oliva, Hinojosa, & Demeshko, 2017)

3.1 Descripción del algoritmo EMO

Birbil y Fang propusieron el algoritmo evolutivo de optimización basado en electromagnetismo, en inglés *electromagnetism-like optimization* (EMO) (Ş. I. Birbil & Fang, 2003), un método creado para resolver problemas de optimización sin restricciones. El algoritmo posee características interesantes, una de ellas es su rápida convergencia a valores óptimos evitando quedar atrapado en soluciones subóptimas comúnmente encontrados en los espacios de búsqueda (De Castro & Von Zuben, 2002; De Jong, 1988; Dorigo, Maniezzo, & Colorni, 1996). Otra característica importante es el buen balance que logra mantener entre resultados obtenidos y el número de evaluaciones necesarias para alcanzar tales resultados. El algoritmo EMO al ser concebido como un método poblacional, mantiene una población $\mathbf{Sp}_t = \{x_{1,t}, x_{2,t}, \dots, x_{N,t}\}$ de N partículas $(x_{i,t})$ con un tamaño d -dimensional $(i=1, 2, \dots, d)$. La población se representa mediante un conjunto factible de soluciones $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^d \mid l_i \leq x \leq u_i\}$

donde t representa la iteración (o generación) actual del algoritmo. Antes de comenzar el proceso iterativo, la población \mathbf{Sp}_t se inicializa tomando muestras uniformemente distribuidas dentro de la región de búsqueda \mathbf{X} . Después de la inicialización de \mathbf{Sp}_t , el proceso iterativo del algoritmo EMO continúa hasta que se alcanza una condición de parada (por ejemplo, un número máximo de iteraciones). Cada iteración del algoritmo EMO consta de dos pasos principales. En el primer paso, cada partícula en \mathbf{Sp}_t se desplaza a una posición distinta usando un mecanismo de atracción y repulsión propio de la teoría del electromagnetismo (Ş. I. Birbil, Fang, & Sheu, 2004). En la segunda etapa, las partículas que han cambiado su posición usando el principio del electromagnetismo son perturbadas localmente mediante el uso de una búsqueda local para finalmente convertirse en miembros de la población \mathbf{Sp}_{t+1} en la siguiente generación. Ambos mecanismos son responsables de llevar las partículas de \mathbf{Sp}_t cerca del óptimo global.

De forma similar a las partículas cargadas bajo la perspectiva de la teoría del electromagnetismo, cada partícula $x_{i,t} \in \mathbf{Sp}_t$ en el espacio de búsqueda \mathbf{X} se considera como una partícula cargada. La carga de cada partícula se calcula de acuerdo con el valor de la función objetivo. Las partículas que tengan un mejor valor descrito por la función objetivo tienen mayores cargas que las partículas restantes. El mecanismo de atracción y repulsión constituye un proceso clave en el algoritmo EMO donde las partículas con más carga atraen otras partículas de \mathbf{Sp}_t mientras que partículas con menos carga repelen otros elementos. Finalmente, se calcula un vector de fuerza total F_i^t ejercida en un punto por la suma de todas las atracciones y repulsiones, y cada $x_{i,t} \in \mathbf{Sp}_t$ se mueve en la dirección de la fuerza total a la posición $y_{i,t}$. Se usa una búsqueda local para explorar la vecindad de cada partícula de acuerdo con su valor de aptitud. Los miembros de la nueva población $x_{i,t+1} \in \mathbf{Sp}_{t+1}$ se determinan usando la siguiente ecuación:

$$x_{i,t+1} = \begin{cases} y_{i,t} & \text{si } f(y_{i,t}) \leq f(z_{i,t}) \\ z_{i,t} & \text{en otro caso} \end{cases} \quad (3.1)$$

En la Ec. (3.1), $z_{i,t}$ es una partícula perturbada en el proceso de búsqueda local. Así, ambos, $y_{i,t}$ y $z_{i,t}$, se evalúan usando la función objetivo $f(\bullet)$, siendo g el argumento de la función. Mediante esta ecuación un elemento de la población se actualiza si el valor de la función objetivo es menor que su valor anterior. Se debe hacer notar que el signo \leq es usado para el caso de la minimización; si se trata de un problema de maximización el símbolo deberá ser invertido.

El algoritmo 3.1 muestra el esquema general de la técnica EMO, en el que se describe cada paso detalladamente.

-
1. Parámetros de entrada: Máximo número de iteraciones $Iter_{\max}$, valores para la búsqueda local como $Iter_{local}$ y el tamaño N de la población.
 2. Inicialización: establecer contador de iteraciones $t=1$, inicializa la población \mathbf{Sp}_t uniformemente en \mathbf{X} e identifica la mejor partícula de \mathbf{Sp}_t .
 3. mientras $t < Iter_{\max}$ hacer
 4. $F_i^t \leftarrow \text{CalcF}(\mathbf{Sp}_t)$
 5. $y_{i,t} \leftarrow \text{Mueve}(x_{i,t}, F_i^t)$
 6. $z_{i,t} \leftarrow \text{Local}(Iter_{local}, \delta, y_{i,t})$
 7. $x_{i,t+1} \leftarrow \text{Selecciona}(\mathbf{Sp}_{t+1}, y_{i,t}, z_{i,t})$
 8. fin mientras
-

Algoritmo 3.1. Pseudocódigo del algoritmo *electromagnetism-like optimization*.

Parámetros de entrada (Línea 1): El algoritmo EMO itera durante $Iter_{\max}$ iteraciones. En la fase de búsqueda local, $n \times Iter_{local}$ es el número máximo de ubicaciones $z_{i,t}$, dentro de una distancia δ de $y_{i,t}$, para cada dimensión i .

Inicialización (Línea 2): Los puntos $x_{i,t}$, $t=1$, se seleccionan uniformemente dentro de \mathbf{X} , por ejemplo $x_{i,1} \approx \text{Unif}(\mathbf{X})$, $i = 1, 2, \dots, N$, donde Unif denota una distribución uniforme dentro de \mathbf{X} . Se calculan los valores de la función objetivo $f(x_{i,t})$. A continuación se selecciona el mejor punto o partícula x_t^B en cada iteración. Para el caso de minimización se aplica la Ec. (3.2) mientras que para maximización es la Ec. (3.3).

$$x_t^B = \arg \min_{x_{i,t} \in \mathbf{Sp}_t} \{f(x_{i,t})\} \quad (3.2)$$

$$x_t^B = \arg \max_{x_{i,t} \in \mathbf{Sp}_t} \{f(x_{i,t})\} \quad (3.3)$$

Calcula fuerza (Línea 4): En este paso, se asigna un valor de carga $q_{i,t}$ a cada punto $x_{i,t}$. La carga $q_{i,t}$ de $x_{i,t}$ depende directamente de $f(x_{i,t})$. Así, los puntos con mejor valor de aptitud reportado por la función objetivo tienen mayor carga que los otros. Las cargas se calculan de acuerdo con la Ec. (3.4).

$$q_{i,t} = \exp \left(-n \frac{f(x_{i,t}) - f(x_t^B)}{\sum_{j=1}^N f(x_{j,t}) - f(x_t^B)} \right) \quad (3.4)$$

En la Ec. (3.4), n es el número de dimensiones del espacio de búsqueda, N es el número de soluciones candidatas mientras que x_t^B es el mejor punto en la iteración actual t .

La fuerza que existe entre dos puntos $x_{i,t}$ y $x_{j,t}$ de la población se calculan usando la Ec (3.5).

$$F_{i,j}^t = \begin{cases} \left(x_{j,t} - x_{i,t} \right) \frac{q_{i,t} \cdot q_{j,t}}{\|x_{j,t} - x_{i,t}\|^2} & \text{si } f(x_{i,t}) > f(x_{j,t}) \\ \left(x_{i,t} - x_{j,t} \right) \frac{q_{i,t} \cdot q_{j,t}}{\|x_{j,t} - x_{i,t}\|^2} & \text{si } f(x_{i,t}) \leq f(x_{j,t}) \end{cases} \quad (3.5)$$

La fuerza total F_i^t , correspondiente a una partícula específica $x_{i,t}$ se calcula ahora como:

$$F_i^t = \sum_{j=1, j \neq i}^N F_{i,j}^t \quad (3.6)$$

Una vez calculadas las fuerzas, el siguiente paso (Línea 5) consiste en mover el punto seleccionado $x_{i,t}$ en la dirección del vector F_i^t usando la siguiente ecuación:

$$y_{i,t} = x_{i,t} + \lambda \frac{F_i^t}{\|F_i^t\|} (RNG), \quad i = 1, 2, \dots, N; \quad i \neq B \quad (3.7)$$

donde $\lambda \approx Unif(0,1)$ es un número aleatorio extraído de una distribución uniforme dentro del intervalo $[0,1]$ para cada coordenada de $x_{i,t}$. RNG denota el rango de movimiento permitido hacia los límites inferiores y superiores correspondientes a cada dimensión. B indica el índice del mejor elemento de la población.

Búsqueda local (Línea 6): Para cada $y_{i,t}$ se genera un máximo $iter_{local}$ de puntos en cada dirección de δ por la vecindad de $y_{i,t}$. Esto implica que el proceso de generación de puntos locales se continua para cada $y_{i,t}$ hasta que se encuentra una mejor solución $z_{i,t}$ o se alcanza el límite $n \times Iter_{local}$ de intentos.

Selección para la siguiente iteración (Línea 7): En este paso se selecciona $x_{i,t+1} \in \mathbf{Sp}_{t+1}$ de entre $y_{i,t}$ and $z_{i,t}$ usando la Ec. (3.1).

El proceso de optimización realizado por el algoritmo EMO involucra diferentes operaciones que son capaces de manejar información local y global del espacio de búsqueda. Este proceso es más complejo que los utilizados por otros enfoques metaheurísticos, los cuales sólo involucran una o dos operaciones matemáticas para modificar su población (Kennedy & Eberhart, 1995; Storn & Price, 1997).

3.2 Segmentación de imágenes usando entropía cruzada.

El cálculo de la entropía cruzada mínima (ECM) para la umbralización de imágenes es una tarea compleja, incluso si se redefine utilizando programación recursiva. El principal problema es obtener los mejores umbrales que minimizan la entropía cruzada. En el caso de realizar una búsqueda exhaustiva se espera una gran carga computacional. En esta sección se describe el uso del algoritmo EMO para la segmentación de imágenes usando como criterio la ECM. Este enfoque usa el algoritmo EMO para encontrar un conjunto de valores de umbral \mathbf{th} óptimos los cuales sean capaces de minimizar la entropía cruzada. En comparación con otros métodos metaheurísticos, el EMO exhibe capacidades de búsqueda interesantes, tales como una rápida convergencia mientras mantiene la habilidad de evitar subóptimos locales en entornos altamente multimodales (De Castro & Von Zuben, 2002; De Jong, 1988; Dorigo et al., 1996). En la literatura relacionada existen estudios que demuestran la capacidad del EMO de generar uno de los mejores balances entre desempeño y la demanda de acceso a funciones objetivo (Birbil et al., 2004; Rocha & Fernandes, 2009; A. ; Wu et al., 2004). Los resultados muestran que este algoritmo puede reducir substancialmente el número de llamadas a la función objetivo mientras preserva sus buenas capacidades de búsqueda. Además de que EMO es una buena alternativa para la optimización global, incluye un proceso para determinar los elementos de una nueva población que involucran varias operaciones descritas en la Ec. (3.1-3.7). A continuación, se describe la implementación de la aplicación para segmentación de imágenes basada en EMO y la ECM.

Para encontrar los valores de umbrales óptimos para la ECM es necesario codificar dichos umbrales dentro de cada partícula del algoritmo EMO. Así, cada partícula usa nt variables de decisión en el algoritmo de optimización. Tales elementos representan los distintos valores de umbrales usados para la segmentación de imágenes. Es por ello por lo que la población del algoritmo EMO será representada de forma:

$$\mathbf{Sp}_i = [\mathbf{th}_1, \mathbf{th}_2, \dots, \mathbf{th}_N], \quad \mathbf{th}_i = [th_1, th_2, \dots, th_{nt}]^T \quad (3.8)$$

donde T hace referencia al operador de transposición, N es el tamaño de la población y nt es el número de umbrales para cada miembro de la población ($i = 1, 2, \dots, N$).

El método de segmentación propuesto se ha implementado considerando la ECM y utilizando como función objetivo la definida en la Ec. (2.28). La implementación del algoritmo EMO con ECM se concreta en los siguientes pasos:

- Paso 1: Leer la imagen I para representarla en escala de grises I_{Gr} .
- Paso 2: Obtener el histograma h^{Gr} de I_{Gr} .
- Paso 3: Inicializar los parámetros del EMO: $Iter_{max}$, $Iter_{local}$, δ , k y N .
- Paso 4: Inicializar la población Sp_i con N partículas aleatorias de nt dimensiones.
- Paso 5: Evaluar Sp_i en la función objetivo de la ECM recursiva $\varphi(Sp_i)$, Ec. (2.29).
- Paso 6: Determinar la carga de cada partícula usando las Ec. (3.4), Ec. (3.5) y (3.6), obteniendo el vector de fuerza total.
- Paso 7: Desplaza a cada individuo de Sp_i en la dirección del vector de fuerza total usando la Ec. (3.7).
- Paso 8: Realizar una búsqueda local en las partículas desplazadas y seleccionar los mejores elementos de acuerdo a su función objetivo.
- Paso 9: El índice t es incrementado en 1, si $t \geq Iter_{max}$ o se alcanza el criterio de parada, finaliza el proceso iterativo y se avanza al paso 10. En caso contrario se retorna al paso 7.
- Paso 10: Seleccionar la mejor partícula x^* de acuerdo con el valor de la función objetivo según Ec. (3.2) y Ec. (2.29).
- Paso 11: Realizar la umbralización de la imagen I_{Gr} con los valores de x^* .

Una vez que EMO encuentra los mejores valores de umbrales, los cuales minimizan la función objetivo, es posible proceder a realizar la umbralización. Para ello, se utiliza la regla definida en la Ec. 2.6.

3.3 Resultados experimentales

En este capítulo se utilizó un conjunto de imágenes de prueba para evaluar el desempeño de la propuesta. El conjunto contiene once imágenes con distintos niveles de complejidad. Todas las imágenes poseen el mismo tamaño (512x512 píxeles). Algunas de estas imágenes son ampliamente utilizadas en la literatura del área (Lena, Cameraman, Hunter, Baboon, etc.) y son extraídas de la base de datos de imágenes USC-SIPI (1981), por tanto estas imágenes permiten una mejor valoración de resultados en términos comparativos, ya que existe una gran cantidad de artículos que las utilizan.

El algoritmo de umbralización multinivel basado en EMO propuesto se compara frente a otras versiones que utilizan distintos algoritmos de optimización como lo es la evolución diferencial (DE), el algoritmo de enjambre de partículas (PSO), búsqueda de armonías (HS), algoritmo de la araña social (SSA) y también la colonia artificial de abejas (ABC).

Todos los métodos han sido programados y evaluados en el entorno de Matlab 8.3 (MathWorks, 2018) utilizando un procesador Xeon E5-2620 CPU @ 2.4Ghz con 16GB de memoria RAM. Dado que las seis implementaciones son estocásticas, se utilizan métricas estadísticas. Para cada experimento cada caso se evalúa un total de 35 veces para cada configuración y con cada imagen. En estas pruebas se utilizan cuatro cantidades de umbrales

siendo $nt = 2,3,4,5$ de acuerdo con (Horng, 2010; Horng & Liou, 2011). La condición de parada se establece en 500 iteraciones.

Los parámetros elegidos para el algoritmo EMO se configuran de acuerdo a los criterios presentados en Birbil y Fang (2003), para facilidad del lector se incluyen en la Tabla 3.1. Tales valores se seleccionan específicamente para el proceso de minimización requerido por ECM.

Tabla 3.1. Parámetros del EMO para ECM.

$Iter_{max}$	$Iter_{local}$	δ	N
200	5	0.25	25

3.3.1 Resultados con entropía cruzada

Esta subsección analiza los resultados de la umbralización obtenidos por el algoritmo EMO usando como función objetivo la ECM (Ec. (2.29)). El enfoque propuesto se aplica al conjunto completo de imágenes de prueba y los resultados de la segmentación se presentan en la Tabla 3.2. Tales valores corresponden a los mejores umbrales encontrados por el algoritmo EMO considerando cuatro valores distintos de umbrales ($nt = 2,3,4,5$). Además, la Tabla 3.2 muestra los valores de métricas estadísticas (PSNR, σ , SSIM y FSIM) obtenidos para cada imagen con el número de umbrales indicado. Observando la Tabla 3.2 es posible notar que los valores obtenidos por EMO son estables (σ) inclusive cuando el número de umbrales se incrementa. Además, la calidad de la imagen segmentada obtenida por EMO se compara con las métricas del PSNR, SSIM y FSIM. Las columnas correspondientes a tales medidas permiten apreciar la capacidad de segmentación del enfoque propuesto de acuerdo con la definición de cada métrica.

Tabla 3.2. Resultados obtenidos por EMO aplicando la ECM sobre el conjunto de imágenes de prueba.

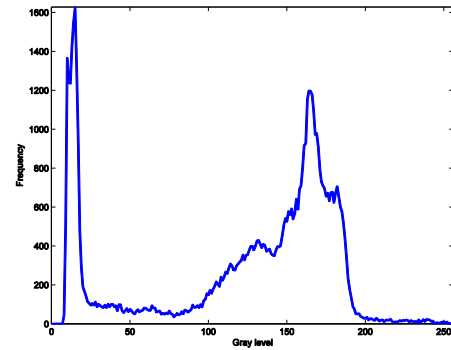
Imagen	nt	Umbrales	PSNR	σ	SSIM	FSIM
Camera man	2	51, 137	18.3128	0.0000	0.8564	0.8100
	3	31, 84, 144	21.6180	3.5466 E-05	0.9003	0.8536
	4	30, 77, 126, 158	23.2970	1.7459 E-04	0.9103	0.8848
	5	28, 70, 114, 145, 171	23.9009	4.1780 E-03	0.9237	0.8968
Lena	2	81, 140	20.9383	3.2483 E-05	0.8612	0.8462
	3	73, 119, 165	21.5202	0.0000	0.8794	0.8426
	4	70, 108, 140, 176	23.4480	7.2898 E-05	0.9006	0.8702
	5	59, 86, 115, 143, 177	24.9227	2.0784 E-04	0.9206	0.8930
Baboon	2	90, 145	21.4273	0.0000	0.9130	0.9155
	3	71, 110, 152	23.1514	1.5435 E-05	0.9431	0.9443
	4	64, 99, 131, 163	23.9436	8.9906 E-05	0.9591	0.9601
	5	56, 84, 111, 138, 168	25.0505	4.4126 E-04	0.9679	0.9676
Hunter	2	26, 90	22.8504	0.0000	0.9008	0.8920
	3	18, 60, 116	22.1675	5.2461 E-04	0.8942	0.8961
	4	14, 43, 84, 131	23.1216	7.4499 E-05	0.9189	0.9216
	5	11, 31, 60, 96, 137	24.3002	1.2439 E-03	0.9392	0.9425
Airplane	2	97, 161	25.4876	2.3546 E-05	0.9624	0.9142
	3	76, 126, 178	24.2310	1.7566 E-04	0.9543	0.9058
	4	63, 105, 147, 190	24.0633	1.5277 E-04	0.9407	0.9029
	5	61, 100, 135, 174, 203	23.6548	1.0091 E-04	0.9395	0.9072
Peppers	2	57, 130	19.4660	0.0000	0.8795	0.8642
	3	45, 88, 138	23.2943	5.5620 E-03	0.9164	0.8858
	4	43, 83, 124, 168	21.9553	0.0000	0.9015	0.8674
	5	42, 80, 111, 142, 175	23.4937	1.4000 E-02	0.9238	0.8929
Living room	2	73, 134	19.4210	0.0000	0.8490	0.8427
	3	44, 95, 145	20.2041	1.0570 E-05	0.8780	0.8760

	4	38, 82, 123, 160	21.4640	1.5015 E-05	0.9101	0.9052
	5	33, 68, 102, 133, 167	23.1660	2.5034 E-03	0.9370	0.9303
Blonde	2	38, 122	18.4613	0.0000	0.8557	0.8598
	3	33, 107, 155	18.0949	0.0000	0.8336	0.8422
	4	31, 96, 132, 164	19.5506	2.5181 E-06	0.8801	0.8881
	5	19, 63, 104, 139, 168	22.9896	9.1403 E-03	0.9339	0.9150
Bridge	2	76, 142	20.4912	2.0776 E-03	0.8705	0.8768
	3	57, 104, 161	20.9807	1.7097 E-03	0.9014	0.9114
	4	49, 88, 129, 181	21.5630	1.1741 E-03	0.9208	0.9229
	5	43, 76, 108, 146, 192	22.5800	3.7022 E-03	0.9414	0.9398
Butterfly	2	88, 138	22.9957	3.5474E-04	0.9064	0.8580
	3	77, 113, 156	23.3959	8.0843E-05	0.9290	0.8802
	4	70, 97, 125, 161	25.3474	9.2064E-05	0.9548	0.9230
	5	66, 88, 111, 135, 165	26.6674	6.6659E-03	0.9681	0.9489
Lake	2	73, 141	24.6248	0.0000	0.9467	0.9305
	3	62, 104, 162	24.2344	1.3607 E-04	0.9523	0.9398
	4	57, 91, 144, 195	20.9612	2.5874 E-04	0.9370	0.9170
	5	73, 141	24.6248	0.0000	0.9467	0.9305

Del conjunto de once imágenes de prueba, se han seleccionado cinco de ellas de acuerdo con su complejidad y peculiaridades presentes en su histograma y con el fin de mostrar los resultados de la segmentación gráficamente. La Figura 3.1 presenta este subconjunto de imágenes y sus respectivos histogramas. Algunos de ellos poseen distribuciones irregulares, particularmente el correspondiente a la Figura 3.1 (j). En este tipo de imágenes los métodos tradicionales se enfrentan a grandes dificultades para obtener los mejores umbrales.



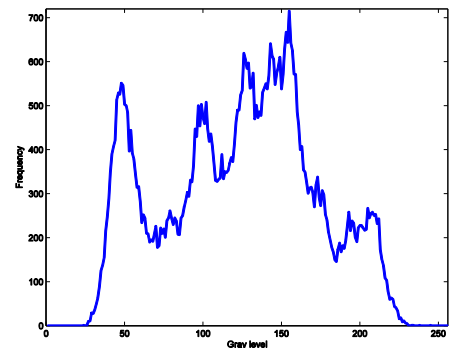
(a)



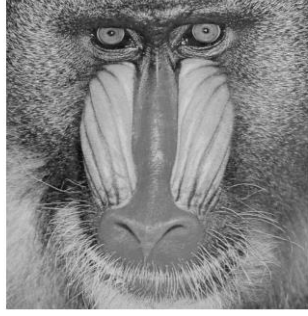
(b)



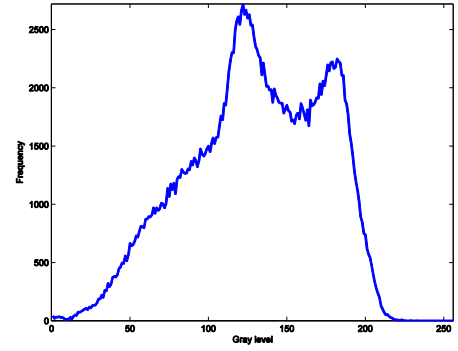
(c)



(d)



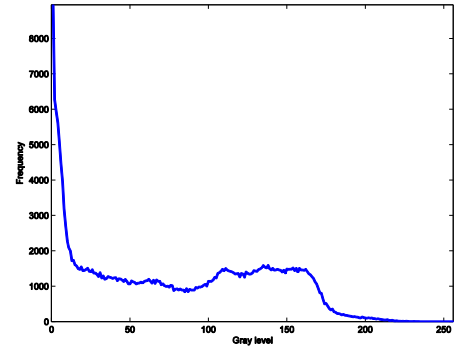
(e)



(f)



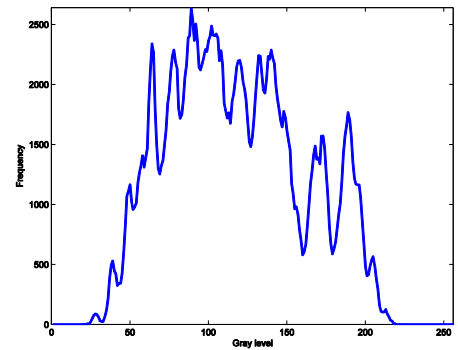
(g)



(h)



(i)

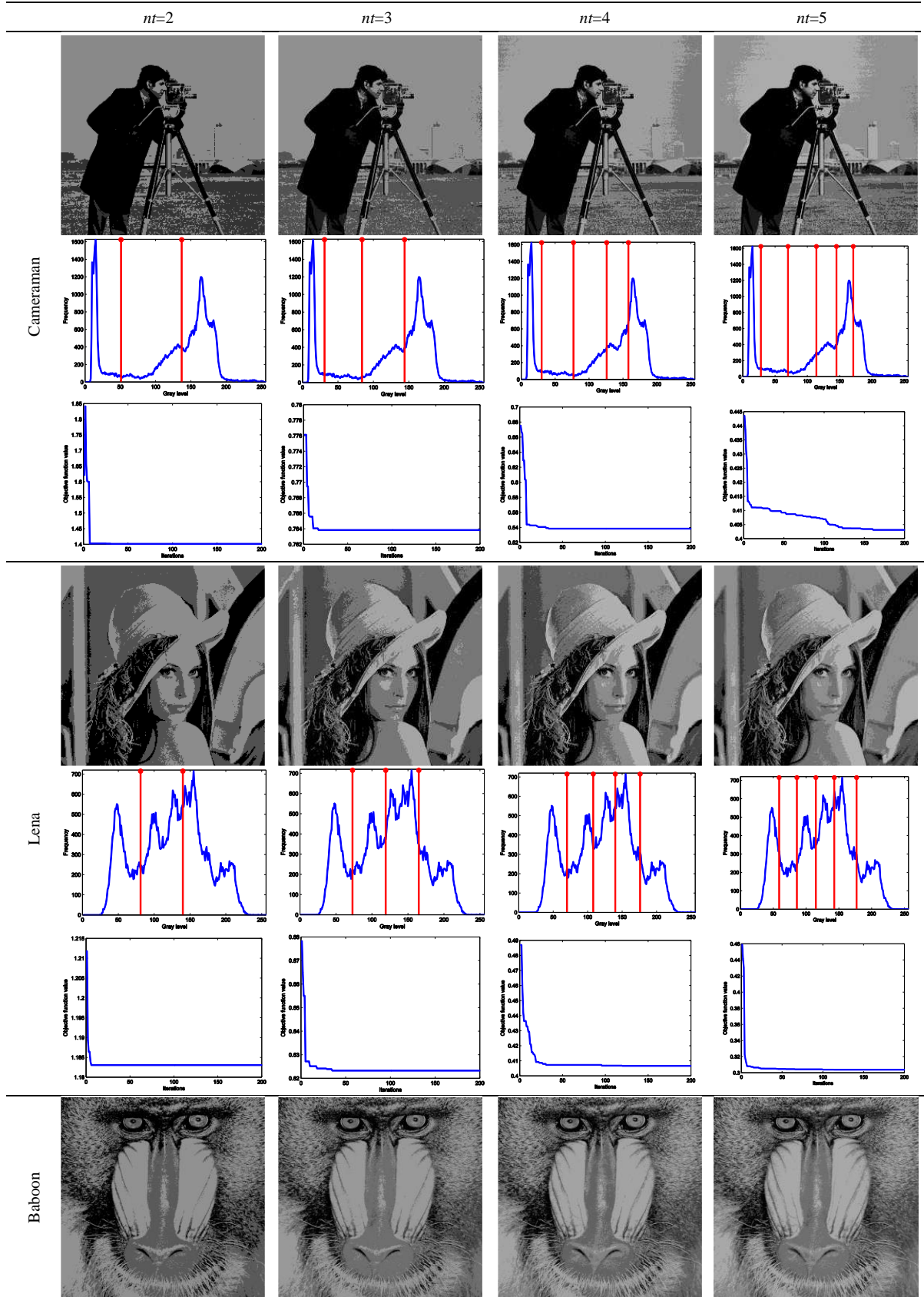


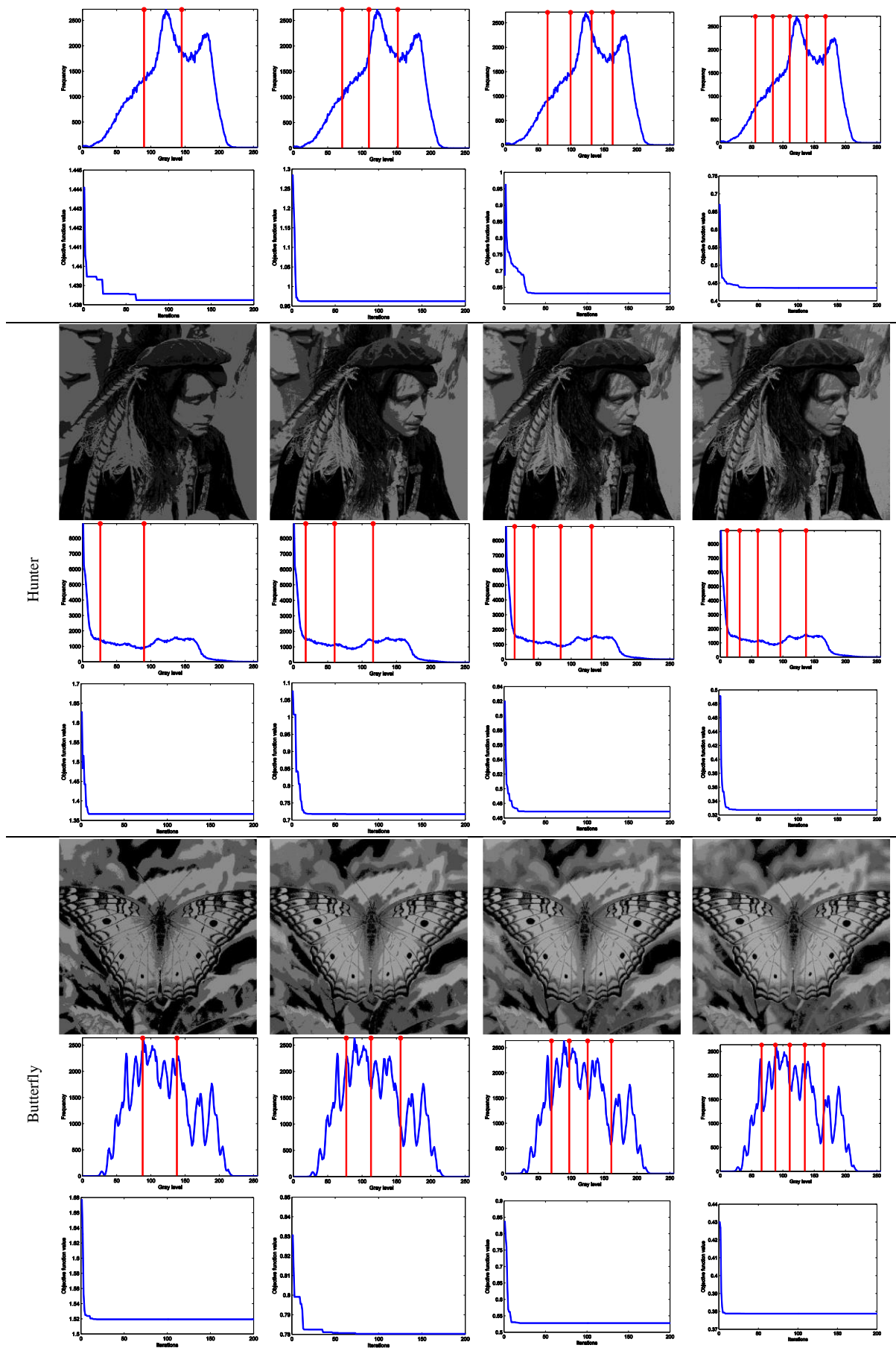
(j)

Figura 3.1. Imágenes usadas para la comparación visual: (a) Cameraman, (c) Lena, (e) Baboon, (g) Hunter y (i) Butterfly. Los histogramas correspondientes a cada imagen (b), (d), (f), (h), (j).

Los resultados de las cinco imágenes seleccionadas se muestran en la Tabla 3.3 considerando cuatro números de umbrales distintos $nt = 2, 3, 4, 5$. Además, en la Tabla 3.3 es posible apreciar que el algoritmo EMO, aplicando la ECM, converge cerca de las primeras 100 iteraciones. Las imágenes segmentadas muestran evidencia de que los resultados suelen ser mejores con números de umbrales $nt = 4$ y $nt = 5$; aun así, es posible observar que la imagen claramente con tres umbrales.

Tabla 3.3. Resultados obtenidos después de aplicar el algoritmo EMO para ECM sobre las imágenes seleccionadas.





3.4 Comparaciones

Para demostrar que el uso del algoritmo EMO aplicado a la ECM es una alternativa interesante para la segmentación de imágenes, el algoritmo propuesto se compara con dos implementaciones del estado del arte. Los métodos empleados para la comparación son los ya mencionados previamente, a saber: el algoritmo evolución diferencial (DE), el algoritmo de enjambre de partículas (PSO), el de búsqueda de armonías (HS), el algoritmo de la araña social (SSA) y colonia artificial de abejas (ABC). Todos estos métodos usan la entropía cruzada mínima como función objetivo. Los seis algoritmos fueron evaluados 35 veces sobre cada imagen seleccionada. Las imágenes utilizadas para esta evaluación son las mismas que en la sección anterior. Para cada imagen I se determinan los valores de PSNR, σ , SSIM, FSIM y la media y desviación estándar de la función objetivo.

Tabla 3.4. Comparación de los valores de media y σ de los algoritmos EMO, DE, PSO, HS, SSA y ABC aplicados sobre el subconjunto de imágenes de prueba usando ECM como función objetivo.

I	nt	EMO		DE		PSO		HS		SSA		ABC	
		σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media
Cameraman	2	0.00 E 00	1.4016	1.99 E-02	1.4168	3.72 E-02	1.428	1.84 E-02	1.4171	2.51 E-02	1.4303	2.05 E-02	1.4244
	3	3.55 E-05	0.7638	4.44 E-02	0.8208	5.84 E-02	0.8352	5.49 E-02	0.8256	6.39 E-02	0.845	6.90 E-02	0.8499
	4	1.74 E-04	0.5385	4.23 E-02	0.6162	4.85 E-02	0.635	4.16 E-02	0.6122	4.97 E-02	0.6317	6.18 E-02	0.6337
	5	4.17 E-03	0.4061	2.79 E-02	0.4792	4.70 E-02	0.4982	3.04 E-02	0.4808	3.53 E-02	0.5009	5.03 E-02	0.5011
Lena	2	3.25 E-05	1.3663	2.17 E-02	1.3829	2.32 E-02	1.3915	1.63 E-02	1.3834	3.58 E-02	1.4001	2.45 E-02	1.3897
	3	0.00 E 00	0.7174	4.89 E-02	0.7853	4.57 E-02	0.793	4.92 E-02	0.7884	5.23 E-02	0.7992	6.78 E-02	0.7927
	4	7.29 E-05	0.4687	4.28 E-02	0.5511	5.11 E-02	0.5751	4.08 E-02	0.5645	5.71 E-02	0.5847	6.72 E-02	0.5731
	5	2.07 E-04	0.3273	3.77 E-02	0.4278	5.78 E-02	0.4486	2.89 E-02	0.4391	4.94 E-02	0.4703	5.57 E-02	0.4695
Baboon	2	0.00 E 00	1.1905	1.65 E-02	1.2096	3.24 E-02	1.2188	1.57 E-02	1.207	2.20 E-02	1.2129	2.14 E-02	1.2157
	3	1.54 E-05	0.7319	3.55 E-02	0.7902	4.31 E-02	0.8014	3.05 E-02	0.7775	4.74 E-02	0.8043	4.99 E-02	0.7973
	4	8.99 E-05	0.5001	4.61 E-02	0.5783	4.09 E-02	0.5961	3.63 E-02	0.5811	4.62 E-02	0.6045	7.31 E-02	0.6113
	5	4.46 E-04	0.3641	3.72 E-02	0.4577	4.64 E-02	0.4772	4.23 E-02	0.4646	3.19 E-02	0.4746	5.20 E-02	0.4808
Hunter	2	0.00 E 00	2.7409	2.59 E-02	2.7653	6.58 E-02	2.8044	2.78 E-02	2.7723	4.11 E-02	2.7946	5.93 E-02	2.7886
	3	5.21 E-04	1.6281	5.80 E-02	1.7308	8.31 E-02	1.7503	5.96 E-02	1.7144	9.82 E-02	1.7862	1.01 E-01	1.7979
	4	7.40 E-05	1.0308	9.54 E-02	1.2322	1.17 E-01	1.2585	1.07 E-02	1.2189	1.17 E-01	1.2866	1.40 E-01	1.2744
	5	1.29 E-03	0.7557	8.15 E-02	0.9689	8.92 E-02	1.0207	8.21 E-02	0.9635	1.00 E-01	1.0154	1.45 E-01	1.0311
Butterfly	2	3.54 E-04	1.1832	6.86 E-03	1.1896	2.12 E-02	1.2046	1.35 E-02	1.1975	1.82 E-02	1.2054	1.75 E-02	1.197
	3	8.08 E-05	0.6233	4.88 E-02	0.6973	6.58 E-02	0.7193	5.71 E-02	0.6941	5.94 E-02	0.7196	4.67 E-02	0.6912
	4	9.21 E-05	0.4065	3.97 E-02	0.5133	6.30 E-02	0.5353	4.81 E-02	0.5128	6.07 E-02	0.5351	6.49 E-02	0.5357
	5	6.66 E-03	0.3076	3.46 E-02	0.4024	4.91 E-02	0.4195	4.58 E-02	0.4016	5.48 E-02	0.4243	6.10 E-02	0.4281

Las comparaciones de los resultados entre los seis métodos se han dividido en dos grupos. En el primero, la Tabla 3.4 muestra los valores de media y σ del valor de la función objetivo determinado por la ECM. En el segundo grupo las Tablas 3.5 y 3.6 presentan valores de métricas de calidad obtenidas después de aplicar los umbrales y realizar la segmentación.

Tales valores reportan evidencia que indica que las imágenes segmentadas con la ECM usando EMO tienen mejor calidad.

Tabla 3.5. Comparaciones de los valores de PSNR, SSIM y FSIM generados por los algoritmos EMO, DE y PSO.

I	nt	EMO			DE			PSO		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Cameraman	2	18.4825	0.8564	0.8139	18.2128	0.8411	0.7986	18.1977	0.845	0.8002
	3	21.7818	0.8994	0.8525	21.4468	0.864	0.8113	21.0884	0.8089	0.8106
	4	23.2188	0.9091	0.8833	22.1301	0.8617	0.8277	22.0455	0.829	0.8288
	5	23.6259	0.9204	0.8932	23.4241	0.8782	0.8383	22.541	0.8141	0.8291
Lena	2	21.234	0.8625	0.8619	20.8154	0.8313	0.8157	20.7911	0.8268	0.8063
	3	21.5202	0.8794	0.8712	21.4249	0.8395	0.794	21.009	0.8467	0.7804
	4	23.5538	0.9001	0.8847	22.4228	0.8628	0.8287	22.9989	0.8589	0.8165
	5	24.8234	0.9203	0.8995	23.5816	0.882	0.8426	23.3937	0.8925	0.8228
Baboon	2	21.9085	0.9149	0.9183	21.4273	0.8879	0.8919	21.3669	0.8724	0.8843
	3	23.0794	0.9425	0.9438	22.6824	0.913	0.9145	22.7909	0.881	0.9043
	4	23.8567	0.9583	0.959	23.4748	0.9201	0.9175	22.9763	0.916	0.9117
	5	24.836	0.9669	0.9669	24.032	0.9211	0.9151	23.7103	0.9131	0.9145
Hunter	2	22.8504	0.9008	0.8976	22.822	0.8757	0.8722	22.7911	0.8521	0.868
	3	22.7167	0.9007	0.8958	22.1533	0.8615	0.8586	21.0875	0.8649	0.8681
	4	23.0168	0.9172	0.9205	22.377	0.884	0.8796	22.6372	0.8591	0.8576
	5	24.1474	0.9376	0.9408	23.3941	0.907	0.9027	22.5532	0.9047	0.899
Butterfly	2	24.0269	0.9055	0.9039	22.9003	0.8885	0.833	22.8248	0.8726	0.8301
	3	23.3668	0.9281	0.9216	22.9889	0.8917	0.8259	23.2509	0.8852	0.8417
	4	25.2881	0.9543	0.938	24.2203	0.9103	0.849	24.1546	0.9005	0.8227
	5	26.2271	0.9655	0.9425	24.5201	0.9217	0.874	24.1146	0.8405	0.7619

Tabla 3.6. Comparación de los valores de PSNR, SSIM y FSIM obtenidos por los algoritmos HS, SSA, y ABC al ser aplicados en la segmentación del subconjunto de imágenes de prueba.

I	nt	HS			SSA			ABC		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Camera man	2	18.2443	0.8445	0.8085	18.2825	0.8459	0.7999	18.2044	0.8542	0.8102
	3	21.5535	0.8870	0.8550	20.7268	0.8725	0.8378	21.5797	0.8848	0.8422
	4	21.9599	0.8721	0.8578	22.7573	0.8959	0.8627	22.2802	0.8913	0.8675
	5	22.9764	0.9020	0.8798	23.0061	0.9001	0.8701	22.1411	0.9106	0.8814
Lena	2	20.9209	0.8450	0.8475	20.9073	0.8506	0.8254	20.8491	0.8511	0.8458
	3	21.0785	0.8533	0.8371	21.2115	0.8617	0.8257	21.2352	0.8532	0.8368
	4	21.3584	0.8572	0.8315	22.5265	0.8892	0.8469	22.3370	0.8627	0.8524
	5	23.1562	0.8925	0.8622	23.1375	0.8986	0.8641	23.2626	0.8795	0.8671
Baboon	2	21.3029	0.8754	0.8940	21.0843	0.8983	0.9201	21.7885	0.9067	0.9089
	3	22.6206	0.9245	0.9358	22.5477	0.9238	0.9350	22.1142	0.9289	0.9308
	4	23.0713	0.9319	0.9444	22.6867	0.9310	0.9327	23.0584	0.9435	0.9464
	5	22.8485	0.9301	0.9366	23.7274	0.9340	0.9446	23.2269	0.9516	0.9516
Hunter	2	22.4354	0.8829	0.8838	21.8970	0.8767	0.8754	21.8218	0.8867	0.8800
	3	22.3198	0.8801	0.9015	21.9116	0.8744	0.8860	21.3750	0.8830	0.8845
	4	22.5711	0.8912	0.9019	22.0705	0.8926	0.9040	22.7938	0.9090	0.9103
	5	22.5337	0.8906	0.9093	22.6585	0.9179	0.9046	22.7033	0.9165	0.9148
Butterfly	2	22.9358	0.8567	0.8722	22.8141	0.8845	0.8544	22.9281	0.9066	0.8567
	3	22.7386	0.8792	0.8595	23.1200	0.8916	0.8741	22.9124	0.9205	0.8709
	4	22.8761	0.9091	0.8722	23.2632	0.8990	0.9017	23.5058	0.9319	0.8904
	5	23.4250	0.9284	0.8875	24.2389	0.9012	0.9077	24.3144	0.9429	0.9095

Los valores de aptitud, obtenidos mediante la evaluación de la función de objetivo se evalúan estadísticamente frente al algoritmo EMO utilizando la prueba de Wilcoxon (sección 2.6.5) la cual se realiza mediante 35 muestras independientes. Para realizar la prueba Wilcoxon es necesario hacer la comparación del algoritmo de interés, en este caso EMO, contra los valores producidos por el resto de los algoritmos que pueden ser considerados como similares. La Tabla 3.7 muestra los valores p producidos por la prueba de Wilcoxon en comparaciones entre los valores de aptitud de los grupos formados por EMO vs. DE, EMO vs. PSO, EMO vs. HS, EMO vs. SSA y EMO vs. ABC. Todos los valores p reportados en la Tabla 3.7 son menores que 0.05 (nivel de significancia 5%) lo que indica una evidencia fuerte en contra de la hipótesis nula, indicando que los valores encontrados por el Algoritmo EMO son diferentes a los de los otros algoritmos.

Tabla 3.7. Valores p de la prueba de Wilcoxon comparando EMO vs. DE, EMO vs. PSO, EMO vs. HS, EMO vs. SSA y EMO vs. ABC.

I	nt	Valores p				
		EMO vs. DE	EMO vs. PSO	EMO vs. HS	EMO vs. SSA	EMO vs. ABC
Camera man	2	1.5291E-14	1.5337E-14	1.5337E-14	1.5328E-14	5.8049E-14
	3	2.3579E-13	2.3579E-13	2.3579E-13	2.3579E-13	2.3579E-13
	4	2.1460E-13	2.1460E-13	2.1460E-13	2.1460E-13	2.1460E-13
	5	5.6572E-13	5.6572E-13	6.1678E-13	5.6572E-13	5.6572E-13
	2	1.3652E-11	2.5260E-13	2.5248E-13	2.3305E-11	2.5198E-13
Lena	3	1.5346E-14	1.5346E-14	1.5346E-14	1.5346E-14	1.5346E-14
	4	2.3579E-13	2.3579E-13	2.3579E-13	2.3579E-13	2.3579E-13
	5	5.3642E-13	5.3642E-13	5.3642E-13	5.3642E-13	5.3642E-13
	2	1.5300E-14	1.5346E-14	1.5337E-14	1.5328E-14	5.8049E-14
	3	2.6616E-13	2.6616E-13	2.6616E-13	2.6616E-13	2.6616E-13
Baboon	4	2.2955E-13	2.2955E-13	2.2955E-13	2.2955E-13	2.2955E-13
	5	5.9456E-13	5.9456E-13	5.9456E-13	5.9456E-13	5.9456E-13
	2	5.7948E-14	5.8015E-14	1.5346E-14	1.5337E-14	5.7914E-14
	3	1.7183E-13	1.7183E-13	1.7183E-13	1.7183E-13	1.7183E-13
	4	4.6150E-13	4.6150E-13	4.6150E-13	4.6150E-13	4.6150E-13
Hunter	5	5.1937E-13	5.1937E-13	5.1937E-13	5.1937E-13	5.1937E-13
	2	2.5210E-14	2.1814E-14	2.6427E-14	2.1840E-14	8.7304E-14
	3	2.4220E-13	2.4220E-13	2.4220E-13	2.4220E-13	2.4220E-13
	4	5.0448E-13	5.0448E-13	5.0448E-13	5.0448E-13	5.0448E-13
	5	3.6185E-13	3.6185E-13	3.6185E-13	3.6185E-13	3.6185E-13
Airplane	2	1.5328E-14	1.5337E-14	5.7847E-14	1.5328E-14	1.5310E-14
	3	1.6535E-12	1.6535E-12	1.6535E-12	3.5421E-12	4.1516E-13
	4	1.5346E-14	1.5346E-14	1.5346E-14	1.5346E-14	1.5346E-14
	5	2.1478E-12	5.0024E-13	2.1478E-12	2.1478E-12	5.0024E-13
	2	1.5346E-14	5.7981E-14	1.5310E-14	1.5328E-14	5.7948E-14
Peppers	3	2.1840E-14	2.1840E-14	2.1840E-14	2.1840E-14	2.1840E-14
	4	1.7580E-13	1.7580E-13	1.7580E-13	1.7580E-13	1.7580E-13
	5	3.0306E-13	3.0306E-13	3.0306E-13	3.0306E-13	3.0306E-13
	2	1.5291E-14	1.5310E-14	1.5310E-14	1.5319E-14	1.5328E-14
	3	1.5346E-14	1.5346E-14	1.5346E-14	1.5346E-14	1.5346E-14
Living room	4	2.1840E-14	2.1840E-14	2.1840E-14	2.1840E-14	2.1840E-14
	5	1.1710E-12	6.4191E-13	7.6272E-13	7.6272E-13	1.2753E-12
	2	9.1839E-14	7.7840E-13	5.6912E-14	4.7117E-14	2.9739E-13
	3	3.6102E-13	2.5310E-13	3.6102E-13	3.0237E-13	3.6102E-13
	4	3.7405E-13	3.7405E-13	3.7405E-13	3.7405E-13	3.7405E-13
Bridge	5	5.6968E-13	5.6968E-13	5.6968E-13	5.6968E-13	5.6968E-13
	2	2.1394E-12	4.4802E-13	1.2753E-12	3.1793E-12	7.5207E-13
	3	2.5273E-13	2.5273E-13	2.5273E-13	2.5273E-13	2.5273E-13
	4	3.0752E-13	3.0752E-13	3.0752E-13	3.0752E-13	3.0752E-13
	5	5.2969E-13	5.2969E-13	5.2969E-13	5.2969E-13	5.2969E-13
Butterfly	2	2.1198E-13	1.5337E-14	1.5337E-14	1.5337E-14	1.5319E-14
	3	4.7746E-13	4.7746E-13	4.7746E-13	4.7746E-13	4.7746E-13
	4	4.2073E-13	4.2073E-13	4.2073E-13	4.2073E-13	4.2073E-13
	5	5.9016E-13	5.9016E-13	5.9016E-13	5.9016E-13	5.9016E-13
	2	2.1198E-13	1.5337E-14	1.5337E-14	1.5337E-14	1.5319E-14
Lake	3	4.7746E-13	4.7746E-13	4.7746E-13	4.7746E-13	4.7746E-13
	4	4.2073E-13	4.2073E-13	4.2073E-13	4.2073E-13	4.2073E-13
	5	5.9016E-13	5.9016E-13	5.9016E-13	5.9016E-13	5.9016E-13

Por otra parte, la Tabla 3.8 presenta una comparación entre el algoritmo EMO y los cinco métodos seleccionados para analizar el tiempo computacional. Estas mediciones se usan para evaluar el esfuerzo computacional requerido por cada algoritmo. Para esta comparación se han realizado 35 experimentos independientes para cada algoritmo. También, el número de iteraciones máximas se establece en 1000 para que cada algoritmo evalúe el conjunto de imágenes de prueba. Dado que los seis algoritmos son de naturaleza estocástica, su comparación en términos de complejidad resulta una tarea difícil, por ello el tiempo computacional resulta una buena herramienta para determinar la rapidez de cálculo de cada método.

Tabla 3.8. Media del tiempo computacional de los algoritmos EMO, DE, PSO, HS, SSA y ABC para la segmentación de imágenes usando la ECM.

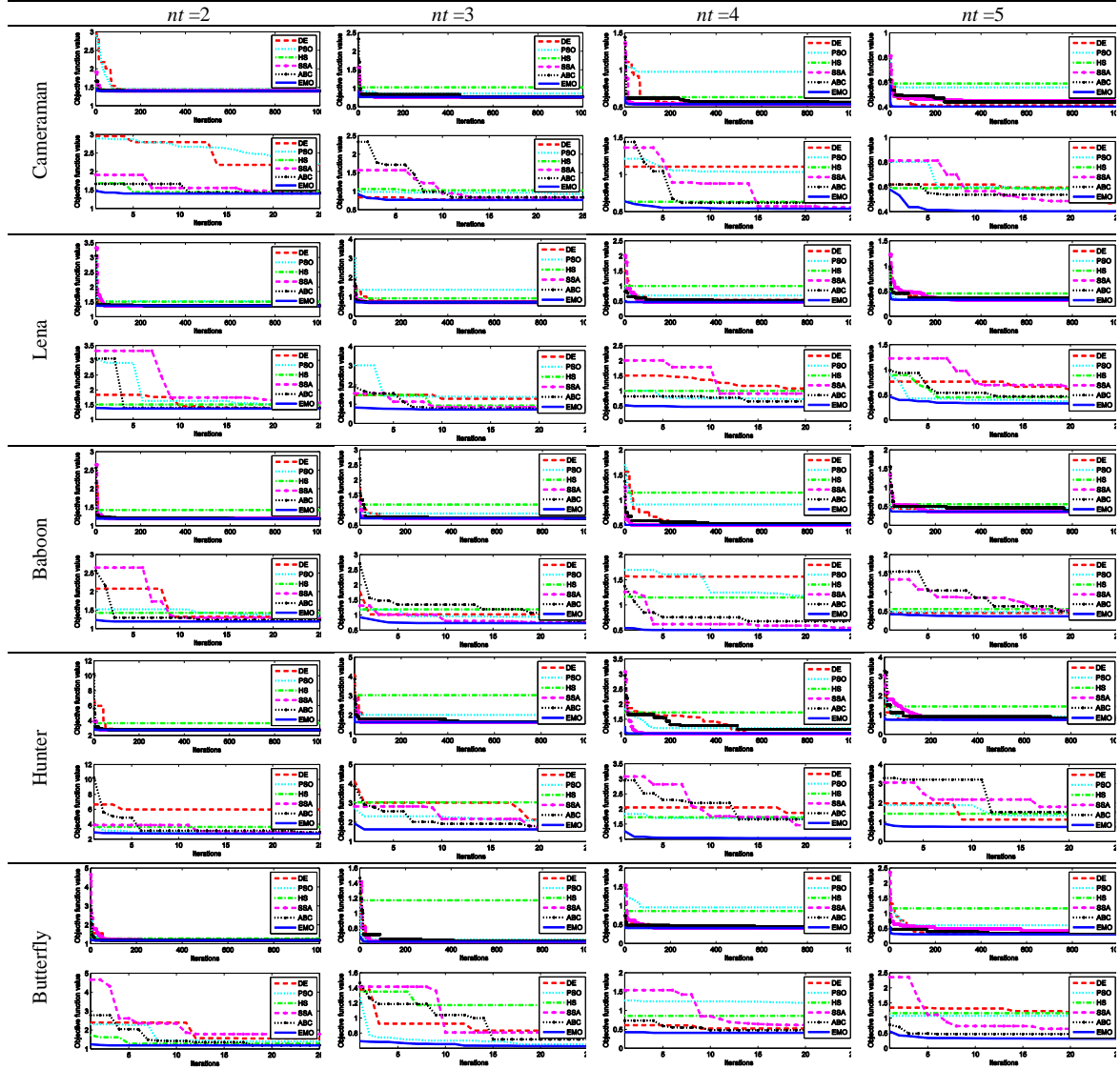
I	nt	Tiempo computacional en segundos					
		EMO	DE	PSO	HS	SSA	ABC
Camera man	2	0.9436	2.8702	2.9128	5.1731	6.6812	4.9803
	3	1.414	3.3423	3.0573	5.6769	7.1879	4.9481
	4	2.5064	3.1482	3.0093	5.3187	7.1748	5.0035
	5	3.7611	3.1084	3.9338	5.843	7.2849	5.0408
Lena	2	1.4837	2.9474	3.0571	4.121	4.7815	5.4233
	3	2.3095	3.1527	3.18	4.6949	4.8725	5.9854
	4	2.8467	3.0956	3.0698	5.1584	4.9822	5.6837
	5	3.6665	3.4252	3.9929	5.6943	5.0831	5.6679
Baboon	2	1.4262	3.1317	3.0253	4.4503	4.7491	5.5001
	3	2.1936	3.1358	3.0467	4.9133	3.8139	5.5502
	4	2.8608	3.1488	3.0035	5.3466	3.5713	5.5536
	5	3.5343	3.5108	3.1198	5.6283	3.6267	5.7071
Hunter	2	1.4975	2.8719	3.0142	4.5676	3.416	5.5245
	3	2.22	2.9633	2.904	4.8718	3.4855	5.6224
	4	3.1627	3.0585	3.0083	5.2737	3.5588	5.8129
	5	3.728	3.2096	3.0446	5.7714	3.9146	5.3861
Airplane	2	1.5541	3.0118	3.0582	4.5824	3.415	5.0549
	3	2.2903	2.9289	2.98	4.8948	3.492	5.3252
	4	2.9445	3.0994	3.1955	5.126	3.5504	5.1442
	5	3.7565	3.2444	3.9576	5.2884	3.9235	5.2147
Peppers	2	1.4794	2.819	2.8964	4.1583	3.4155	5.0975
	3	2.2575	2.7814	3.0897	4.922	3.4946	5.1322
	4	3.0144	2.8531	3.0865	5.3589	3.5559	5.2255
	5	3.7579	3.2648	3.0161	5.7062	3.6226	5.2086
Living room	2	1.6184	2.8374	2.9206	4.4757	3.4112	5.0795
	3	2.2417	2.9434	2.8998	4.9375	3.4766	5.083
	4	3.041	3.0548	3.0273	5.3077	3.5398	5.1982
	5	3.7145	3.1756	3.8089	5.3819	3.9101	5.3334
Blonde	2	1.5284	2.8266	2.712	4.4955	3.4527	5.0428
	3	2.2144	2.9556	2.7727	4.9646	3.4924	5.1213
	4	2.9952	3.0334	2.6926	5.3587	3.55	5.2055

	5	3.7327	3.3047	3.7712	5.6642	3.9076	5.2187
	2	1.4589	2.8325	2.783	4.5453	3.4253	5.1206
	3	2.2875	2.8466	2.9345	4.9387	3.4903	5.2781
Bridge	4	2.991	3.8041	3.0259	5.4024	3.577	5.3119
	5	3.674	3.8382	2.965	5.6041	3.6278	5.4317
	2	1.5039	3.0215	2.9142	4.7109	5.2115	5.0502
	3	2.2411	2.9894	2.9877	5.0698	5.386	4.9276
Butterfly	4	3.0091	3.2101	3.8212	5.3102	5.615	5.0205
	5	3.7002	3.3017	3.3976	5.7413	5.8124	5.0661
	2	1.4726	2.8825	3.4073	4.5335	5.1208	4.8969
	3	2.1217	2.947	3.5072	4.8877	5.2051	4.9079
Lake	4	2.9959	3.2373	3.4629	5.1438	5.5869	5.0462
	5	3.5776	3.2751	3.7326	5.4053	5.871	5.5951

En la Tabla 3.8 es posible observar que el algoritmo basado en EMO en la mayoría de los casos necesita menos tiempo para encontrar una solución óptima. Además, cuando el número de umbrales aumenta, por ejemplo, $nt=5$, el algoritmo EMO requiere más tiempo que DE, PSO o SSA. Por otro lado, el algoritmo HS requiere más tiempo dado que utiliza una única partícula durante el proceso iterativo. Mientras tanto, los operadores del ABC son totalmente iterativos ya que utiliza dos poblaciones para encontrar las mejores soluciones. El algoritmo SSA tiene configuraciones iniciales que afectan a su eficacia y lo hacen difícil de configurar para que tenga un buen rendimiento. Basado en esos hechos, el tiempo computacional es mayor en comparación al de EMO, DE o PSO, pero es cercano a los valores de HS y SSA.

En la Tabla 3.9 se presentan los valores de aptitud obtenidos en cinco de las imágenes del subconjunto de prueba. Para este experimento cada algoritmo se evalúa durante 1000 iteraciones guardando en cada iteración el mejor valor encontrado. Para tener un mejor entendimiento y apreciación de la convergencia en la Tabla 3.9 se incluye una ampliación de la gráfica de los valores de aptitud. La ampliación considera sólo 25 iteraciones de cada uno de los seis algoritmos. Este número se ha elegido experimentalmente, ya que se observó que los valores de ECM disminuyen después de las 25 iteraciones en todos los casos. A partir de la Tabla 3.9 es posible deducir que el algoritmo propuesto basado en EMO requiere menos iteraciones en comparación con métodos similares. En este contexto el criterio de parada podría modificarse. Inclusive, los resultados presentados en el caso de HS refuerzan la observación previa donde se esperaba que el algoritmo HS requiriera más iteraciones para converger dado que es un algoritmo basado en trayectoria. Finalmente, es posible establecer que el enfoque propuesto basado en EMO y la ECM alcanza valores de entropía cruzada mínimos con menos iteraciones, obteniendo resultados mejores que los que producen DE, PSO, HS, SSA y ABC.

Tabla 3.9. Comparación la evolución de los valores de aptitud de DE (línea roja), PSO (línea cian), HS (línea verde), SSA (línea magenta), ABC (línea negra) and EMO (línea azul).



3.5 Resumen

En este capítulo se presentó la implementación del algoritmo EMO usando la entropía cruzada mínima para la segmentación de imágenes. La propuesta se comparó frente a algoritmos clásicos y del estado del arte tales como DE, PSO, HS, SSA y ABC, que han sido analizados convenientemente. Los resultados indican que el algoritmo EMO es capaz de converger a soluciones óptimas con pocas iteraciones. Además, es posible observar un incremento en la calidad de la segmentación de las imágenes usando EMO y ECM.

4 Umbralización de imágenes especiales

En este capítulo se analiza la factibilidad del uso de EAs para la segmentación de imágenes especiales. En este contexto, el término especial se usa para aquellas imágenes que no son capturadas con sensores comunes. El primer caso de interés es la segmentación de resonancias magnéticas cerebrales donde se propone una nueva metodología para su umbralización (Oliva, Hinojosa, Cuevas, et al., 2017) mientras que en el segundo caso se evalúa el desempeño de EAs tradicionales para la umbralización de imágenes térmicas (Hinojosa, Pajares, Cuevas, & Ortega-Sanchez, 2018). En la sección 4.1 se analizan las imágenes de resonancia magnética, describiendo los métodos aplicados, junto con la base experimental desarrollada y el análisis de los resultados. En la sección 4.2 se hace lo propio para las imágenes térmicas.

4.1 Resonancias magnéticas

La segmentación de las resonancias magnéticas (RM) cerebrales ha atraído la atención de los investigadores debido a su gran número de aplicaciones. Las imágenes por RM se utilizan para el diagnóstico médico de tumores y otras enfermedades neurológicas, tales como la esclerosis múltiple, la demencia y la esquizofrenia (Kamber et al., 1995). Comúnmente, el análisis de las imágenes cerebrales de RM se realiza por expertos mediante evaluaciones visuales basadas en sus experiencias y habilidades. Sin embargo, la inspección visual humana es limitada y conlleva mucho tiempo. Estas restricciones han conducido al desarrollo de técnicas asistidas por ordenador destinadas a extraer las estructuras anatómicas (Moeskops et al., 2016;

Ortiz et al., 2013; Suzuki & Toriwaki, 1991), en las que muchas metodologías se basan en la segmentación.

En esta sección, se introduce un nuevo método para la segmentación de imágenes. El enfoque, llamado ECM-CSA, integra el criterio de umbralización por entropía cruzada mínima (ECM) con la búsqueda del cuervo CSA. Bajo el método propuesto, el algoritmo CSA se utiliza para minimizar la entropía cruzada entre las clases generadas por los umbrales. Por lo tanto, en cada generación, CSA codifica un conjunto de umbrales candidatos en una solución. La función objetivo evalúa la entropía cruzada determinando la calidad de la solución propuesta. Guiados por los valores de la función objetivo, se generan nuevas soluciones candidatas utilizando los operadores predefinidos por CSA mientras que la calidad de la segmentación mejora a medida que el proceso evoluciona. Aunque el método propuesto puede ser utilizado como un algoritmo de segmentación genérico, sus características permiten obtener excelentes resultados en la segmentación automática de imágenes complejas de RM. En tales circunstancias, este enfoque ha sido evaluado usando dos conjuntos de imágenes de referencia; el primer conjunto está compuesto de imágenes generales comúnmente usadas en la literatura del procesamiento de imágenes, mientras que el segundo conjunto corresponde a imágenes cerebrales de RM. Para evaluar el rendimiento de ECM-CSA, sus resultados se comparan con los producidos por dos técnicas de optimización metaheurística populares como la evolución diferencial (DE) y la búsqueda de armonías (HS) (Loganathan, 2001; Rainer Storn & Price, 1997).

4.1.1 Descripción del algoritmo CSA

El algoritmo de búsqueda del cuervo (CSA) propuesto por Askarzadeh (2016), es un algoritmo evolutivo inspirado en el comportamiento inteligente de los cuervos. Los cuervos son considerados entre los animales más inteligentes del mundo (Black, 2013). Por lo tanto, sus comportamientos pueden proporcionar heurísticas interesantes. CSA se inspira en el comportamiento de robo que muestran los cuervos. Dicho comportamiento puede resumirse de la siguiente manera: los cuervos memorizan la posición en la que esconden la comida sobrante, un cuervo puede seguir a otro para robar en sus escondites, con el fin de proteger sus escondites de comida, los cuervos pueden engañar a los ladrones moviéndose al azar.

Desde un punto de vista computacional, en el algoritmo CSA la población $\mathbf{C}^k \left(\{ \mathbf{c}_1^k, \mathbf{c}_2^k, \dots, \mathbf{c}_N^k \} \right)$ de N individuos (cuervos) evoluciona desde un punto inicial ($k=0$) hasta un número total de generaciones ($k = gen$). Cada cuervo \mathbf{c}_i^k ($i \in [1, \dots, N]$) representa un vector d -dimensional $\{ \mathbf{c}_{i,1}^k, \mathbf{c}_{i,2}^k, \dots, \mathbf{c}_{i,N}^k \}$ donde cada dimensión corresponde a una variable de decisión del problema a resolver. En el algoritmo CSA, una nueva población \mathbf{C}^{k+1} considera dos posibles estados: el primero es cuando el cuervo en cuestión conoce que está siendo seguido para ser

robado mientras que en el segundo caso el cuervo desconoce este hecho. Para modelar este comportamiento un factor de probabilidad de conocimiento AP_i^k determina el estado a seleccionar. Cada nuevo elemento de la población se genera como:

$$\mathbf{c}_i^{k+1} = \begin{cases} \mathbf{c}_i^k + r_i \times fl \times (\mathbf{m}_j^k - \mathbf{c}_i^k) & r_j \geq AP_i^k \\ \text{posición aleatoria} & \text{en otro caso} \end{cases} \quad (4.1)$$

donde r_i y r_j son números aleatorios extraídos de una distribución uniforme entre 0 y 1, fl es el parámetro que controla la longitud del vuelo y \mathbf{m}_j^k es la memoria del cuervo j en la iteración k , es decir, su escondite de comida. CSA sólo requiere la configuración de dos parámetros, fl y AP_i^k . Además, la ecuación de actualización es simple, por lo que se pueden generar implementaciones amigables y poderosas.

La primera etapa es la inicialización del problema (dimensiones y límites) y los parámetros de la CSA (fl , AP , criterios de parada). El siguiente paso consiste en inicializar aleatoriamente las posiciones de los cuervos y evaluarlas usando la función objetivo. Las posiciones se distribuyen uniformemente en el espacio de búsqueda. Después de esto, las nuevas posiciones se generan de acuerdo con la Ec. 4.1 y se verifica su factibilidad. Todas las nuevas posiciones se evalúan en la función objetivo y se actualiza la memoria. Finalmente, se verifica el criterio de parada para terminar o continuar con el proceso iterativo. Los criterios de parada dependen de la implementación de la CSA, pero se suelen utilizar dos reglas comunes: 1) el uso de un número predefinido de generaciones máximas y 2) cuando el valor de la función objetivo de los mejores cuervos converge y por tanto no cambia a lo largo de las iteraciones.

4.1.2 Umbralización mediante CSA y entropía cruzada

En este capítulo, se presenta una nueva metodología combinada llamada ECM-CSA para la umbralización de imágenes. La umbralización multinivel usando entropía cruzada divide la imagen en un número finito de clases mediante la determinación de valores umbral, donde cada nuevo umbral añade complejidad al problema aumentando las restricciones y la modalidad del espacio de búsqueda, especialmente cuando el histograma de la imagen presenta irregularidades. El algoritmo CSA se utiliza para minimizar la entropía cruzada entre clases. Para cada generación, CSA codifica un conjunto de valores de umbral candidatos en una solución. La función objetivo utiliza la entropía cruzada para determinar la calidad de la solución propuesta. Siguiendo las reglas de la CSA y el valor de la función objetivo, se generan nuevas soluciones candidatas utilizando los operadores predefinidos de CSA a la vez que se mejora la calidad de la segmentación a medida que evoluciona el proceso.

Dado que la umbralización multinivel puede tratarse como un problema de optimización, la función objetivo para el criterio de entropía cruzada se indica como:

$$\begin{aligned} \arg \min_{\mathbf{th}} \quad & f_{Cross}(\mathbf{th}) \\ \text{sujeto a} \quad & \mathbf{th} \in \mathbf{X} \end{aligned} \quad (4.2)$$

donde $f_{Cross}(\mathbf{th})$ es la función de entropía cruzada, Ec. (2.28), y $\mathbf{X} = \{th \in \mathbb{R}^{nt} | 0 \leq th_i \leq 255, i=1,2,\dots,nt\}$ es la región factible, limitada por los valores de intensidad del intervalo 0-255. De esta forma, el algoritmo CSA se utiliza para encontrar los valores de umbrales que satisfacen la Ec. 2.28.

El método ECM-CSA propuesto aprovecha la metodología fácil de usar propuesta por Askarzadeh (2016) donde una sola ecuación proporciona suficientes capacidades de búsqueda. Además, la mayoría de EAs requieren de un paso de ajuste de parámetros que consume mucho tiempo, mientras que CSA sólo requiere la selección de los dos parámetros indicados previamente y llamados probabilidad de conocimiento AP y longitud de vuelo fl . La naturaleza no codiciosa de CSA fomenta diversas soluciones a través del espacio de búsqueda. Para asegurar la convergencia del algoritmo, cada cuervo retiene en su memoria como caché de alimento la mejor solución encontrada por ese cuervo en particular. La mezcla de estos mecanismos proporciona un algoritmo de búsqueda eficiente y robusto. ECM-CSA también se beneficia de la definición propuesta por Li (1993) de entropía cruzada mínima, que es más rápida y precisa que los enfoques tradicionales como la varianza entre clases de Otsu (1979). Sin embargo, la reciente publicación de CSA proporciona sólo un puñado de publicaciones que utilizan y evalúan dicho algoritmo. Por lo tanto, el potencial real y las limitaciones de la metodología requieren un análisis exhaustivo. En este trabajo, se observó que el parámetro fl no modifica drásticamente el rendimiento general del algoritmo, mientras que la selección correcta del valor AP mejora significativamente los resultados. Para su funcionamiento cada umbral \mathbf{th} se incorpora dentro del algoritmo de optimización como una variable de decisión dentro de un elemento de la población. Así, la población se representa como:

$$\mathbf{Sp}_t = [\mathbf{th}_1, \mathbf{th}_2, \dots, \mathbf{th}_N], \mathbf{th}_i = [th_{i1}, th_{i2}, \dots, th_{int}]^T \quad (4.3)$$

donde t es el número de la iteración, N es el tamaño de la población, T hace referencia al operador de transposición y nt es el número de umbrales aplicados a la imagen.

En esta investigación, la implementación de ECM-CSA considera el uso de la entropía cruzada como función objetivo y se detalla paso a paso de la forma siguiente:

Paso 1: Adquirir imagen I y guardarla como la matriz \mathbf{I}_{Gr} .

Paso 2: Calcular el histograma h^{Gr} de \mathbf{I}_{Gr} .

- Paso 3: Inicializar parámetros de CSA: probabilidad de conocimiento AP_i^k y longitud de vuelo fl .
- Paso 4: Inicializar población \mathbf{Sp}_t de N soluciones aleatorias con nt dimensiones cada una.
- Paso 5: Evaluar la función objetivo (f_{Cross}) para cada elemento \mathbf{Sp}_t . Ec. 2.29.
- Paso 6: Generar número aleatorio r_j .
- Paso 7: Si $r_j \geq AP$ genera nueva solución mediante $\mathbf{c}_i^{k+1} = \mathbf{c}_i^k + r_j \times fl \times (\mathbf{m}_j^k - \mathbf{c}_i^k)$.
- Paso 8: Si $r_j < AP$ genera nueva solución mediante $\mathbf{c}_i^{k+1} =$ posición aleatoria.
- Paso 9: Verificar si los N nuevos cuervos son factibles.
- Paso 10: Si el criterio de parada no ha sido satisfecho saltar al paso 5.
- Paso 11: Generar la imagen segmentada \mathbf{I}_s con la mejor solución \mathbf{g} y la Ec. 2.6.

4.1.3 Configuración experimental.

La metodología propuesta ha sido validada usando dos conjuntos de imágenes de referencia; el primer conjunto está compuesto de imágenes generales comúnmente usadas en la literatura de procesamiento de imágenes, mientras que el segundo conjunto está formado por imágenes cerebrales de RM. El primer conjunto de imágenes de referencia se utiliza para analizar las capacidades de búsqueda de CSA. Para ello, se utiliza un marco de trabajo de alta dimensión para demostrar las capacidades de búsqueda del algoritmo. La mayor parte de la literatura relacionada se centra en la búsqueda de valores umbral óptimos en un esquema de hasta 5 dimensiones (5 valores umbral), mientras que, en este planteamiento, la búsqueda se lleva a cabo en un espacio de hasta 32 dimensiones. El número de umbrales seleccionados $nt=2^n, n \in \{1, 2, \dots, 5\}$ para el primer conjunto de imágenes de referencia proporciona un rango de comparación más amplio. La mayoría de los trabajos relacionados sólo tienen en cuenta los resultados de imágenes segmentadas con 2, 3, 4 y 5 umbrales, lo que lleva a una diferencia bastante pequeña entre los enfoques comparados. La superioridad de un algoritmo frente a otros enfoques en cuanto a calidad y velocidad se hace evidente a medida que aumenta el número de umbrales. Se establece el límite de 32 umbrales ya que no son muchas las aplicaciones que requieren un mayor número de clases. Este conjunto está compuesto por diez imágenes de referencia de propósito general ampliamente utilizadas en la literatura de procesamiento de imágenes para probar diferentes métodos (Cameraman, Lena, Peppers, Baboon, entre otros). Todas las imágenes tienen la misma resolución espacial, es decir el mismo tamaño (512 x 512 píxeles) y están almacenadas en formato TIFF.

El segundo conjunto de imágenes de referencia se utiliza para determinar el rendimiento de ECM-CSA cuando se aplica específicamente a las imágenes cerebrales de RM y está compuesto por un grupo de 8 imágenes extraídas de la base de datos Brainweb (2018). Debido a la naturaleza de las imágenes del segundo conjunto de pruebas es posible establecer un número aproximado de clases presentes en las imágenes. Contrariamente al primer conjunto de imágenes de referencia donde las imágenes variaban de personas, paisajes y objetos, el segundo conjunto de referencia sólo contiene imágenes cerebrales donde el número de clases no varía. Esta característica permite centrar la búsqueda considerando un máximo de cinco umbrales $nt = 2, 3, 4, 5$.

Para ambos conjuntos de referencia, ECM-CSA obtuvo resultados que mostraban una calidad y consistencia de segmentación competitiva. Además, los resultados se validan en un marco estadísticamente significativo. Todos los experimentos se realizaron usando Matlab 8.3 (MathWorks, 2018) en un procesador con una CPU i5-4210 a 2.3Ghz con 6GB de RAM.

Cada experimento de un algoritmo basado en la población, como CSA y DE, tiene un criterio de parada establecido en 2500 iteraciones, pero el algoritmo se termina si el valor de *fitness* no mejora durante el 10% del criterio de parada. Dado que el HS se basa en una sola partícula, su criterio de parada se fija en el tamaño de la población de CSA y DE multiplicado por 2500. Utilizando este criterio, el algoritmo HS puede encontrar valores óptimos.

4.1.4 Resultados experimentales

Los parámetros utilizados para la implementación de CSA son los propuestos en su definición (Askarzadeh, 2016). DE utiliza los parámetros presentados en Pedersen (2010) mientras que HS utiliza lo indicado en Oliva, Cuevas, & Pajares (2013) para una aplicación similar. La Tabla 4.1 muestra todos los parámetros para cada algoritmo empleados en los experimentos.

Tabla 4.1. Parámetros seleccionados para CSA, DE y HS.

CSA	DE	HS
Población N: 20	Población N: 20	Memoria de armonías: 100
Probabilidad de conocimiento AP:0.1	Probabilidad de cruce: 0.7455	Proporción de consideración: 0.75
Longitud del vuelo fl:2	Peso diferencial: 0.9362	Velocidad de la afinación: 0.5
		Distancia de ancho de banda 0.5
		Número de improvisaciones:300

4.1.4.1 Evaluación del algoritmo CSA aplicado a imágenes estándar

En esta subsección se analizan los resultados proporcionados por las implementaciones de ECM basadas en CSA, DE y HS, después de haber sido aplicadas para segmentar las primeras imágenes de referencia. La distancia entre distribuciones, también conocida como

entropía cruzada f_{Cross} , se selecciona como función objetivo. La metodología propuesta se aplica al conjunto completo de imágenes de referencia.

La Tabla 4.2 muestra los mejores valores de umbral obtenidos después de probar CSA, DE y HS con f_{Cross} como función objetivo, evaluando cinco números diferentes de umbrales $nt \in [2, 4, 8, 16, 32]$. Dado que la mayoría de los algoritmos metaheurísticos funcionan bien en espacios de búsqueda con pocas dimensiones, se analiza el comportamiento de ECM-CSA en un marco de altas dimensiones ampliando la búsqueda hasta 32 dimensiones (32 umbrales). Los valores umbral obtenidos para la mayoría de las imágenes con $th=2,4$ muestran evidencia de que los tres algoritmos seleccionados (CSA, DE y HS) tienden a converger a los mismos valores umbral. A medida que aumenta el número de umbrales, los resultados se van distanciando debido a la alta dimensionalidad del espacio de búsqueda del problema. Bajo tales circunstancias, los algoritmos pueden ser intensamente evaluados en un entorno multimodal.

Tabla 4.2. Mejores umbrales encontrados por CSA, DE y HS.

Imagen	nt	CSA	DE	HS
Cameraman	2	50 137	50 137	50 137
	4	29 77 126 158	29 76 125 157	29 76 125 157
	8	13 25 50 83 114 136 156 176	26 49 80 113 140 155 175 189	22 47 81 112 134 154 172 201
	16	11 15 21 32 46 63 82 99 113 125 137 149 160 170 180 205	10 13 18 30 52 81 103 124 129 142 152 162 166 172 184 213	11 15 21 33 47 63 82 101 115 127 140 152 162 171 181 207
	32	4 8 11 13 16 20 25 31 38 46 54 62 71 81 94 104 109 116 121 127 134 140 148 154 160 166 172 177 183 193 202 226	6 7 11 11 14 19 26 29 41 47 64 75 79 102 112 116 119 121 129 134 138 145 154 160 166 177 182 188 188 212 213 223	5 9 12 16 19 28 38 46 53 64 72 83 95 105 110 117 124 128 136 147 148 157 160 166 171 179 180 186 193 205 233 247
Lena	2	83 142	83 142	83 142
	4	71 109 141 177	71 109 141 177	71 109 141 177
	8	52 69 90 111 130 147 166 191	54 70 94 117 137 152 164 196	50 66 86 105 123 142 163 189
	16	43 51 60 70 81 92 102 112 122 132 141 151 161 173 188 204	42 52 64 70 76 92 104 118 130 134 149 155 169 190 195 235	43 50 58 68 79 91 102 113 125 136 146 156 166 178 191 206
	32	37 42 46 50 54 59 64 69 75 81 86 92 98 104 109 114 120 126 131 136 141 146 152 158 163 169 175 183 191 199 208 216	10 24 30 43 49 56 65 74 85 91 93 97 102 103 109 115 128 130 139 149 152 157 159 160 164 172 178 192 195 208 232 240	11 15 38 46 48 54 62 68 74 86 93 99 105 109 114 125 129 139 146 147 152 157 163 173 175 184 193 205 212 212 229 241
Living room	2	73 134	74 134	73 134
	4	38 83 124 161	38 83 124 161	38 83 124 161
	8	21 43 66 89 111 131 152 181	30 43 60 90 108 127 150 184	21 42 64 86 108 129 151 180
	16	10 20 33 46 59 72 86 98 110 120 132 142 152 165 181 202	13 21 41 54 73 76 90 104 108 117 129 142 158 167 183 203	10 21 34 47 59 72 85 99 110 121 131 141 151 163 180 203
	32	2 6 11 16 23 30 36 42 50 56 63 68 75 83 90 96 102 108 115 121 126 131 136 142 147 155 162 170 180 190 201 215	3 5 7 13 25 33 35 42 44 51 66 81 88 97 100 106 112 113 121 123 131 142 148 150 152 162 164 180 193 198 207 208	5 11 18 26 29 40 46 50 59 62 69 72 79 88 101 107 114 124 130 133 141 147 153 161 164 172 177 188 201 216 219 244
Blonde	2	38 122	40 123	38 122
	4	31 96 132 165	31 96 132 165	31 96 132 165
	8	11 49 77 95 117 138 157 177	9 46 80 100 114 131 157 179	11 49 77 95 117 138 157 177
	16	5 22 42 59 74 83 93 104 116 128 138 148 158 168 178 188	3 22 25 40 51 69 88 101 111 121 132 137 150 165 182 191	5 20 39 59 74 83 94 107 120 132 143 152 161 170 180 189
	32	2 5 12 20 31 40 50 59 68 74 80 85 90 95 101 106 113 120 125 130 136 141 146 150 156 161 167 173 179 186 191 199	4 14 28 46 51 53 61 61 68 75 82 86 91 92 102 108 110 112 119 128 132 134 135 138 144 154 162 167 175 184 194 223	5 11 19 36 51 59 71 73 79 80 83 86 95 103 117 122 124 131 140 145 152 158 163 169 177 182 186 190 193 226 236 240
Peppers	2	54 127	54 127	54 127
	4	37 77 119 165	37 77 119 165	37 77 119 165
	8	22 43 68 89 109 134 158 183	23 42 68 88 106 130 156 182	22 43 68 89 109 133 158 183
	16	14 22 32 45 58 71 83 93 104 117 131 145 158 171 185 200	11 19 32 36 58 73 89 99 112 124 141 160 172 178 197 200	15 24 34 47 60 73 85 95 106 119 134 147 160 173 186 200
	32	7 12 16 21 26 32 40 48 54 61 68 75 81 87 93 99 105 113 121 128 135 141 147 152 158 164 170 178 186 192 199 209	14 18 22 29 41 46 49 59 68 73 79 88 93 103 112 120 123 128 131 132 145 154 165 175 185 193 212 222 232 238 241 256	7 13 19 26 35 39 50 55 57 68 71 80 86 94 95 99 102 113 123 133 141 152 158 159 168 178 183 190 201 210 228 244

Woman Dark Hair	2	84 156	84 157	84 156
	4	62 92 133 185	62 92 133 185	62 92 133 185
	8	53 67 86 110 134 158 185 215	52 67 87 106 133 160 183 213	53 66 85 108 132 156 183 214
	16	48 54 61 69 79 90 101 114 127	50 54 61 69 80 91 98 112 127 131	47 51 57 64 73 84 95 108 121
		140 153 167 182 197 213 231	144 159 183 205 231 239	135 149 164 180 196 214 231
	32	42 47 50 54 58 62 67 72 77 81	9 47 52 54 60 61 66 68 75 83 89	14 34 44 47 49 51 54 59 65 66
Splash		88 94 99 105 113 119 125 133	93 93 101 106 118 120 121 125	72 77 83 91 95 99 110 120 131
		140 146 153 161 168 176 183	130 135 147 156 170 181 181 195	142 145 158 165 172 184 190
		190 197 204 213 222 228 240	207 214 218 224 235	196 208 216 223 235 250
	2	68 144	67 142	68 144
	4	35 80 127 186	35 80 127 186	35 80 127 186
	8	26 48 67 86 103 127 159 201	25 51 68 89 102 128 159 208	25 47 66 85 102 126 159 201
Baboon	16	19 32 44 55 65 76 86 95 104	23 32 41 44 58 72 85 101 114 117	20 33 45 55 65 76 86 96 105 113
		111 121 136 155 175 195 219	131 142 164 174 202 203	124 140 157 177 198 221
	32	15 20 26 33 39 44 49 55 61 67	16 21 33 39 40 43 44 46 53 62 65	15 20 27 33 41 47 53 59 65 67
		73 79 84 89 93 97 101 105 109	74 79 88 94 98 100 102 111 118	71 78 85 86 94 98 104 110 116
		113 118 124 132 141 150 157	136 145 168 173 174 179 188 188	119 131 133 135 157 168 176
		166 177 188 203 217 229	196 216 224 236	196 201 218 228 240 255
Jet	2	92 144	92 144	92 144
	4	66 100 130 161	66 100 130 161	66 100 130 161
	8	46 67 86 105 122 139 158 178	49 65 86 102 116 134 151 173	46 67 86 105 123 140 159 179
	16	29 44 56 67 78 89 99 109 118	43 56 58 69 84 90 97 104 114 131	28 42 54 66 77 88 99 109 119
		127 136 146 157 168 179 191	142 151 163 174 178 181	128 138 148 159 170 180 191
	32	14 24 31 38 44 50 56 62 68 73	24 39 41 57 60 70 78 79 88 92	18 26 35 39 50 56 67 76 82 84
Sailboat on lake		79 85 91 96 100 105 111 117	102 107 113 119 127 127 131 135	91 100 106 113 113 123 128 131
		122 127 131 136 140 146 153	136 145 156 163 164 172 175 185	135 144 153 158 168 178 185
		158 165 171 178 184 191 200	194 205 215 221 226 227	188 192 200 205 234 239 240
	2	97 162	97 162	97 162
	4	64 107 150 193	64 107 150 193	64 107 150 193
	8	54 84 106 128 153 179 198 211	53 87 105 126 150 177 197 213	53 81 101 121 145 172 194 209
Sailboat on lake	16	38 55 71 83 95 106 117 129 143	18 35 62 74 88 98 108 114 126	36 50 65 79 91 102 113 125 139
		158 171 183 194 202 209 216	145 159 176 178 190 207 220	154 169 183 194 202 209 216
	32	3 28 36 45 54 62 70 77 83 90 97	3 34 39 40 41 52 62 71 80 89 94	29 32 40 48 49 57 63 71 79 87
		103 109 115 122 129 136 143	98 111 114 126 127 139 145 146	89 95 100 112 122 132 143 146
		151 159 166 173 179 185 190	153 159 171 172 188 195 197 201	157 158 168 174 185 190 197
		194 199 203 207 211 215 220	203 207 212 218 250	203 205 207 212 216 220 253
Sailboat on lake	2	74 142	74 142	74 142
	4	57 91 143 195	57 91 143 195	56 89 142 195
	8	40 54 69 89 114 144 173 202	43 62 76 96 125 146 176 204	39 53 68 88 113 143 173 202
	16	30 39 47 55 64 74 85 98 113	27 42 49 56 63 80 84 101 121 140	28 37 45 52 60 69 80 93 108 125
		130 148 165 179 190 202 217	160 173 186 209 228 240	143 161 176 189 202 216
	32	1 12 23 29 35 40 44 48 52 57 62	15 26 34 43 47 51 58 61 64 65 76	12 25 34 39 45 49 52 59 61 68
Sailboat on lake		67 73 80 87 94 101 109 117 128	85 92 93 104 114 132 135 144	71 82 93 94 101 113 126 137
		138 148 156 164 172 179 185	147 155 160 170 176 183 189 198	144 146 153 161 169 180 186
		192 200 207 215 223	201 220 230 235 246	195 207 209 210 221 226 243

En la Tabla 4.3 se presentan datos estadísticos como la media y la desviación estándar (σ). La media reportada se obtiene a partir de un experimento de 35 ejecuciones que evalúa como función objetivo la entropía cruzada definida en la Ec. (2.29). Dado que se trata de un problema de minimización, se espera que la media sea lo más baja posible. De los tres algoritmos implementados (CSA, DE y HS), CSA es capaz de encontrar los mejores valores umbral que minimizan la función objetivo.

Tabla 4.3. Valores de la función objetivo de las distintas implementaciones de ECM para el primer conjunto de imágenes de prueba generales.

Image	nt	CSA		DE		HS	
		Media	σ	Media	σ	Media	σ
Cameraman	2	1.4446	0.0025	1.4647	0.0183	1.5202	0.0059
	4	0.5629	0.0032	0.6007	0.0262	0.5901	0.0006
	8	0.2238	0.0028	0.2942	0.0155	0.2394	0.0051
	16	0.0722	0.0032	0.1280	0.0111	0.0817	0.0115
	32	0.0233	0.0012	0.0502	0.0039	0.0347	0.0012
Lena	2	1.3503	0.0025	1.3690	0.0164	1.4212	0.0048
	4	0.4708	0.0037	0.5078	0.0231	0.4989	0.0057
	8	0.1584	0.0019	0.2264	0.0207	0.1670	0.0004
	16	0.0520	0.0031	0.0986	0.0096	0.0549	0.0017
	32	0.0163	0.0011	0.0396	0.0035	0.0288	0.0016

Livingroom	2	1.8770	0.0021	1.9003	0.0196	1.9951	0.0154
	4	0.7665	0.0040	0.8110	0.0305	0.8092	0.0068
	8	0.2610	0.0029	0.3561	0.0216	0.2778	0.0061
	16	0.0794	0.0021	0.1429	0.0101	0.0953	0.0038
	32	0.0235	0.0011	0.0537	0.0028	0.0393	0.0025
Blonde	2	1.5224	0.0017	1.5343	0.011	1.6025	0.0034
	4	0.5366	0.0044	0.5699	0.0206	0.5652	0.0013
	8	0.1719	0.0027	0.2438	0.0201	0.1813	0.0033
	16	0.0521	0.0030	0.0993	0.0088	0.0526	0.0013
	32	0.0160	0.0010	0.0372	0.0031	0.0254	0.0001
Peppers	2	1.8621	0.0040	1.8898	0.0204	1.9569	0.0068
	4	0.7161	0.0054	0.7781	0.0384	0.7553	0.0008
	8	0.2551	0.0049	0.3501	0.0245	0.2691	0.0036
	16	0.0783	0.0032	0.1450	0.0119	0.0837	0.0018
	32	0.0242	0.0012	0.0572	0.0046	0.0415	0.0032
Woman Dark Hair	2	1.4233	0.0037	1.4417	0.0149	1.4952	0.0056
	4	0.5266	0.0038	0.5615	0.0217	0.5556	0.0036
	8	0.1680	0.0028	0.24	0.0197	0.1753	0.0024
	16	0.0569	0.0041	0.0981	0.0076	0.0552	0.0005
	32	0.0185	0.0013	0.0383	0.0037	0.0261	0.001
Splash	2	2.2751	0.0028	2.298	0.0201	2.3923	0.0005
	4	0.8114	0.0031	0.8664	0.0378	0.8533	0.0030
	8	0.2188	0.0035	0.3089	0.0241	0.2332	0.0044
	16	0.0668	0.0028	0.1263	0.0096	0.0702	0.0028
	32	0.0211	0.0011	0.0462	0.0031	0.0336	0.0006
Baboon	2	1.0885	0.0027	1.1072	0.0175	1.14460	0.0005
	4	0.4452	0.0039	0.4861	0.0280	0.47187	0.0076
	8	0.1585	0.0022	0.2288	0.0200	0.16863	0.0027
	16	0.0514	0.0030	0.0987	0.0070	0.05302	0.0010
	32	0.0159	0.0009	0.0382	0.0029	0.02908	0.0006
Jet	2	0.8164	0.0010	0.822	0.0054	0.86205	0.0067
	4	0.3382	0.0019	0.3555	0.0124	0.35847	0.0017
	8	0.1096	0.0018	0.1511	0.0126	0.11760	0.0051
	16	0.0343	0.0021	0.0637	0.0050	0.03507	0.0021
	32	0.0110	0.0006	0.0244	0.0018	0.01764	0.0005
Sailboat on lake	2	1.4304	0.0029	1.4448	0.0133	1.50937	0.0103
	4	0.6343	0.0031	0.6700	0.0232	0.66507	0.0036
	8	0.2060	0.0029	0.2841	0.0257	0.21420	0.0015
	16	0.0657	0.0024	0.1196	0.0098	0.0716	0.0059
	32	0.0209	0.0009	0.0459	0.0031	0.0347	0.0021

En la sección 2.6, se introdujeron tres métricas de calidad, que se reproducen aquí por comodidad: *Peak Signal Noise Ratio* (PSNR), *Structural Similarity Index* (SSIM) y *Feature Similarity Index* (FSIM). El objetivo de estas métricas es evaluar la calidad de la imagen segmentada comparándola con la imagen original. Para las tres métricas, un valor alto indica una representación precisa de la imagen original.

Con el fin de establecer un modelo de referencia para comparar los resultados del enfoque propuesto, se implementa y evalúa el método clásico de Otsu (1979) para todo el conjunto de imágenes de referencia con tres algoritmos metaheurísticos. Dado que la función de entropía cruzada se minimiza, mientras que la varianza entre clases se maximiza, los valores de *fitness* no pueden compararse directamente. Para superar este problema, se contrasta la calidad de las imágenes segmentadas de cada enfoque con respecto al PSNR. La Tabla 4.4 muestra el PSNR generado por cada experimento.

Tabla 4.4. Comparación de los valores de PSNR obtenidos por CSA, DE y HS para entropía cruzada y el método de Otsu.

Image	nt	Entropía cruzada			Método de Otsu		
		CSA	DE	HS	CSA	DE	HS
Cameraman	2	15.9977	16.0037	15.5178	15.0362	14.6311	14.5556
	4	21.4231	21.2830	20.7776	20.6406	21.0243	20.6871
	8	25.9145	25.1436	24.8912	25.0075	25.2883	25.1535

	16	31.4143	29.0534	30.4845	30.8975	30.3384	31.0723
	32	37.1093	33.5438	34.5856	35.5327	35.3635	35.5755
Lena	2	15.6372	15.6366	15.1681	12.5669	12.6007	12.7133
	4	18.7857	18.7711	18.2182	17.4860	17.3036	17.4378
	8	23.3044	22.6425	22.6562	21.3257	21.4849	22.0128
	16	29.6854	26.8279	28.7519	28.4563	27.9376	27.8872
	32	37.2611	32.9528	33.2069	31.6590	29.4250	28.4686
Livingroom	2	16.3067	16.3067	15.8168	14.3946	13.7776	13.8624
	4	20.4330	20.4016	19.8056	18.3707	18.4126	18.3967
	8	25.5823	24.6698	24.9327	24.9349	24.8798	24.9689
	16	31.1539	28.9843	30.2537	30.9721	30.4887	31.0989
	32	36.6436	33.2211	33.7799	35.1318	34.7629	35.0233
Blonde	2	14.7983	14.796	14.3576	12.5935	12.2782	12.8562
	4	19.2065	19.1929	18.6226	19.1572	19.1915	19.1572
	8	26.4012	25.1298	25.6400	24.6503	24.6777	25.5613
	16	32.5018	29.8263	31.7121	29.2286	29.5547	29.112
	32	38.2956	34.1692	34.6468	28.9646	29.5456	28.7032
Peppers	2	15.4950	15.4396	14.9817	12.0578	11.7910	11.9489
	4	20.2982	20.1879	19.6741	18.6608	18.5082	18.6425
	8	25.4441	24.5885	24.6655	24.5579	24.3902	24.5958
	16	30.9490	28.2993	30.0746	28.4325	29.4725	29.0455
	32	36.7179	32.6577	33.3549	32.7740	32.3887	32.605
Woman Dark Hair	2	14.0313	14.0322	13.6104	12.0203	12.0733	12.2958
	4	16.8290	16.8103	16.3435	15.6595	15.5805	15.6781
	8	19.7058	19.2465	19.1943	19.1826	18.4021	18.7115
	16	24.2454	24.2081	23.1113	23.6972	23.0617	25.8992
	32	35.4863	30.7002	31.2013	26.3284	27.5433	31.0324
Splash	2	15.3878	15.3841	14.9262	9.72490	9.53860	9.72690
	4	19.4305	19.3874	18.8455	18.3250	17.0876	17.2665
	8	25.1993	24.3335	24.4594	24.5988	24.2083	24.7422
	16	30.6315	28.3716	29.7898	29.2106	29.2523	29.0773
	32	35.9632	33.0156	33.3499	33.5241	33.0466	33.4219
Baboon	2	15.8757	15.8774	15.3994	10.9911	10.8118	11.0492
	4	20.9663	20.8338	20.3373	17.9615	17.8907	17.9778
	8	26.6284	25.3969	25.8505	24.9576	24.1832	24.9269
	16	32.4899	29.5991	31.4009	29.4283	29.2898	28.9531
	32	38.2307	34.0164	34.3016	32.9882	32.6924	32.8527
Jet	2	14.8736	14.8741	14.4274	12.5989	11.4996	11.9582
	4	21.2358	21.0921	20.5938	19.2776	19.3855	19.3045
	8	26.7901	25.6144	25.8937	25.4857	25.2497	25.2142
	16	32.4417	29.8470	31.5194	29.9132	29.3420	29.6200
	32	38.8118	34.7042	35.2961	29.6218	31.2678	32.9738
Sailboat on lake	2	14.2893	14.2771	13.8606	12.1359	11.7854	12.8466
	4	17.8982	17.9428	17.3603	15.8598	15.8959	15.8691
	8	24.3622	23.3960	23.5948	22.7627	22.2225	22.7197
	16	30.8055	27.9829	29.9086	28.4566	28.7639	28.4901
	32	36.7543	32.9337	33.7124	33.2817	32.1775	33.5117

Los algoritmos metaheurísticos son en general, procesos complejos con varios elementos y operaciones estocásticas. En tales condiciones, es difícil realizar un análisis de complejidad desde una perspectiva determinista. Por lo tanto, el tiempo computacional es comúnmente usado para evaluar su esfuerzo computacional. Para evaluar este esfuerzo, se realiza un experimento donde el tiempo es medido y comparado para cada algoritmo cuando operan sobre un conjunto de imágenes representativas. La Figura 4.1 muestra el tiempo computacional requerido para la evaluación tanto de la entropía cruzada como del método de Otsu (1979) usando CSA como algoritmo de optimización. En el eje horizontal, los nombres del conjunto de imágenes de referencia de propósito general se enumeran con grupos de cinco barras para cada imagen. Cada barra representa el tiempo necesario para evaluar la CSA con la técnica y número de umbrales correspondiente. Las barras están agrupadas por el número de umbrales para 2, 4, 8, 16 y 32 para cada imagen de izquierda a derecha. Se puede notar

fácilmente que la entropía cruzada requiere menos tiempo de cálculo en comparación con el método de Otsu, especialmente para un alto número de umbrales.

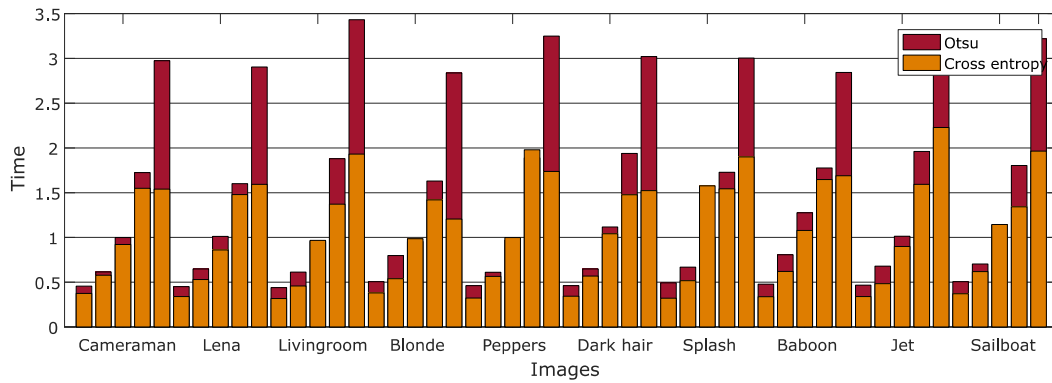


Figura 4.1. Comparación de tiempo Otsu vs. Entropía cruzada usando CSA.

La Tabla 4.5 muestra los valores de SSIM, y FSIM de la evaluación de las imágenes segmentadas. Como se describe en la sección 4.4, las imágenes segmentadas se generan utilizando la Ec. 2.6 y los valores umbral calculados por un algoritmo metaheurístico. En este caso, se utilizan CSA, DE y HS considerando la entropía cruzada como función objetivo. Debe notarse que los resultados reportados son promediados después de un experimento de 35 evaluaciones. CSA supera a DE y HS en las tres métricas de calidad.

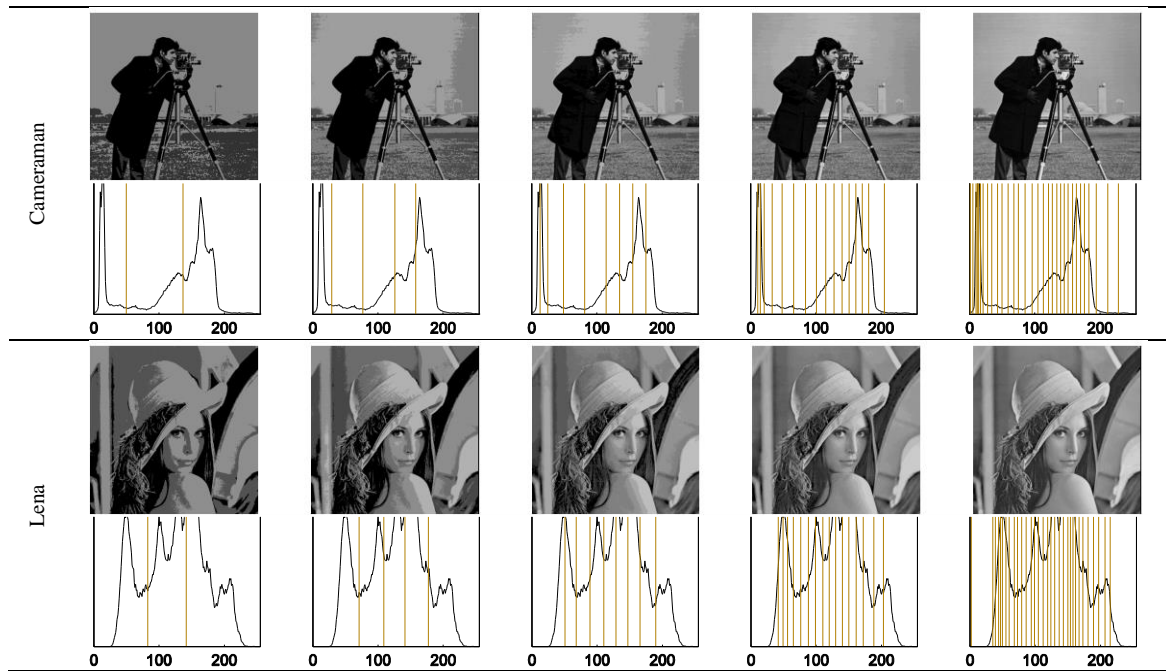
Tabla 4.5. Comparación de los valores de SSIM y FSIM para el conjunto de imágenes generales de referencia.

Imagen	nt	CSA		DE		HS	
		SSIM	FSIM	SSIM	FSIM	SSIM	FSIM
Cameraman	2	0.5891	0.7951	0.5891	0.7952	0.589	0.7951
	4	0.6688	0.8849	0.6685	0.8829	0.6699	0.8851
	8	0.7635	0.9485	0.744	0.9373	0.7616	0.946
	16	0.9148	0.9771	0.8578	0.9632	0.8909	0.9759
	32	0.9716	0.9944	0.9414	0.9835	0.9602	0.9904
Lena	2	0.5601	0.7674	0.5599	0.7674	0.56	0.7674
	4	0.6499	0.8549	0.65	0.8546	0.6497	0.8549
	8	0.7809	0.9087	0.764	0.9014	0.7819	0.9079
	16	0.9034	0.9661	0.8542	0.9396	0.9008	0.9668
	32	0.9727	0.9948	0.9345	0.9815	0.9486	0.9867
Livingroom	2	0.5454	0.7581	0.5454	0.7581	0.5454	0.7581
	4	0.7147	0.882	0.7137	0.8808	0.7147	0.8819
	8	0.8512	0.9559	0.8278	0.9422	0.8501	0.9583
	16	0.9405	0.988	0.9073	0.9759	0.9395	0.9892
	32	0.9789	0.9963	0.9536	0.9899	0.9669	0.9942
Blonde	2	0.5907	0.7144	0.5904	0.7145	0.5885	0.7156
	4	0.7055	0.8743	0.7053	0.8731	0.7047	0.8743
	8	0.8436	0.9589	0.8266	0.9427	0.8441	0.9591
	16	0.9408	0.9912	0.9046	0.9757	0.9433	0.9918
	32	0.9795	0.9971	0.9517	0.9896	0.9647	0.994
Peppers	2	0.5891	0.7522	0.5891	0.7522	0.5891	0.7522
	4	0.6716	0.849	0.6716	0.8471	0.6722	0.849
	8	0.7875	0.9328	0.7728	0.921	0.7873	0.9326
	16	0.9048	0.9822	0.8532	0.9597	0.9037	0.9825
	32	0.9686	0.9954	0.9252	0.984	0.9451	0.9907
Woman Dark Hair	2	0.3353	0.7166	0.3354	0.7167	0.3353	0.7166
	4	0.4493	0.7873	0.4479	0.786	0.4506	0.7867
	8	0.5805	0.8564	0.5589	0.8409	0.5857	0.8554
	16	0.7745	0.9153	0.7614	0.9006	0.7654	0.9033
	32	0.9511	0.985	0.8957	0.9578	0.9137	0.966
Splash	2	0.5772	0.7524	0.5774	0.7526	0.5772	0.7524
	4	0.6975	0.84	0.6969	0.8384	0.6975	0.8393

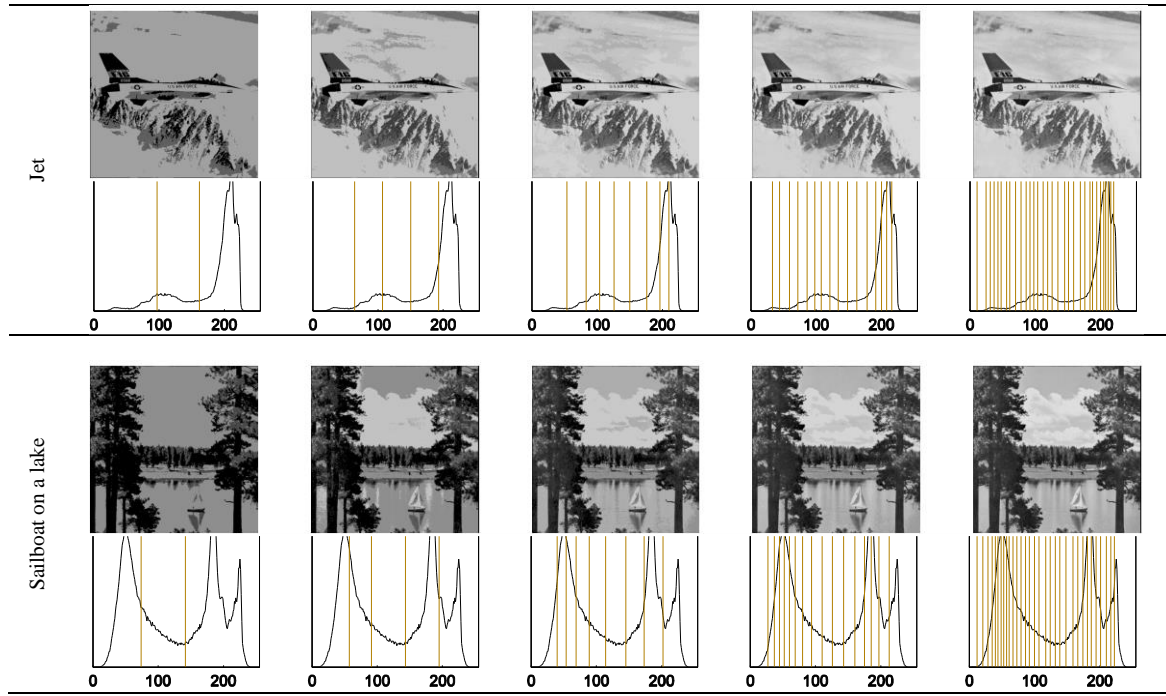
	8	0.7909	0.9169	0.7736	0.9039	0.7902	0.9167
	16	0.8875	0.9702	0.8477	0.9438	0.8837	0.9696
	32	0.9543	0.9855	0.9187	0.9727	0.932	0.9786
Baboon	2	0.6035	0.8174	0.6036	0.8174	0.6035	0.8174
	4	0.7844	0.906	0.7797	0.9045	0.7844	0.906
	8	0.9028	0.9698	0.8792	0.956	0.902	0.9688
	16	0.9665	0.9937	0.9372	0.9821	0.9655	0.9937
	32	0.9892	0.9986	0.9709	0.9937	0.9786	0.9966
Jet	2	0.746	0.8051	0.746	0.8051	0.746	0.8051
	4	0.8061	0.888	0.8033	0.8862	0.8065	0.8881
	8	0.8686	0.963	0.8519	0.9455	0.8675	0.9617
	16	0.9462	0.9885	0.9138	0.9728	0.9442	0.9884
	32	0.9822	0.9972	0.9583	0.9902	0.9694	0.9937
Sailboat on lake	2	0.5281	0.835	0.5279	0.8347	0.5281	0.835
	4	0.6293	0.882	0.6306	0.8819	0.6288	0.8821
	8	0.8287	0.9466	0.8011	0.9364	0.8274	0.9461
	16	0.9336	0.986	0.8933	0.9719	0.9343	0.9862
	32	0.9772	0.9969	0.9493	0.9894	0.9644	0.9938

La Tabla 4.6 presenta los resultados visuales después de aplicar el ECM-CSA a todo el conjunto de imágenes de referencia de propósito general. Para cada imagen, la primera fila presenta la imagen segmentada. Las imágenes fueron segmentadas usando la Ec. 2.6 y los mejores valores de umbral encontrados por el ECM-CSA. De izquierda a derecha se utilizó un número de umbrales de 2, 4, 8, 16 y 32 para demostrar las capacidades de búsqueda de la CSA en un espacio de búsqueda de altas dimensiones. La segunda fila presenta el histograma de la imagen y sus correspondientes valores umbral.

Tabla 4.6. Resultados visuales de la metodología ECM-CSA sobre el conjunto de imágenes de referencia general.







4.1.4.2 Evaluación del algoritmo CSA aplicado a RM.

Esta subsección analiza los resultados proporcionados por las implementaciones de ECM basadas en CSA, DE, y HS, después de usarse para segmentar las imágenes médicas de referencia. El ECM-CSA se aplica al conjunto completo de imágenes de referencia, las cuales se extraen de los cortes de la RM con valores de 1, 2, 5, 10, 36, 72, 108 y 144 para el eje z. Estos valores se seleccionan para adquirir imágenes representativas de diferentes secciones del cerebro. Para las imágenes cerebrales por RM los umbrales son: $nt \in [2, 3, 4, 5]$.

Tabla 4.7. Resultados de ECM-CSA aplicado al conjunto de imágenes de RM cerebrales.

Imagen	nt	CSA		DE		HS	
		Media	σ	Media	σ	Media	σ
Z1	2	3.4002	0	3.454	0.0422	3.4209	0.0309
	3	1.6976	0	1.7636	0.0401	1.7201	0.0142
	4	0.9924	0	1.0929	0.061	1.0142	0.014
	5	0.7055	0.0068	0.8297	0.0692	0.7287	0.0105
Z2	2	3.3614	0	3.4037	0.0296	3.3783	0.0219
	3	1.6694	0	1.7382	0.0408	1.6947	0.0181
	4	0.9882	0	1.0736	0.0472	1.0106	0.0107
	5	0.7054	0.002	0.825	0.0713	0.726	0.009
Z5	2	3.4281	0	3.4595	0.0269	3.4482	0.0206
	3	1.5929	0	1.6708	0.0527	1.6203	0.0142
	4	0.9991	0	1.0856	0.0369	1.0312	0.0469
	5	0.7004	0.0089	0.8103	0.0502	0.7305	0.015
Z10	2	3.3689	0	3.4244	0.0364	3.4009	0.0307
	3	1.6002	0	1.6981	0.0574	1.6233	0.0149
	4	1.1008	0	1.2047	0.0476	1.1232	0.0287
	5	0.7062	0	0.8439	0.0594	0.7637	0.0463
Z36	2	3.2285	0	3.2796	0.0337	3.2572	0.028
	3	1.7369	0.0042	1.8092	0.0429	1.761	0.0131
	4	1.1412	0	1.2263	0.0413	1.1697	0.0311
	5	0.6853	0	0.8386	0.0737	0.7326	0.0227
Z72	2	2.0598	0	2.1231	0.0584	2.079	0.0262
	3	1.1428	0	1.2526	0.0714	1.1658	0.022
	4	0.6606	0	0.7989	0.0737	0.7078	0.029
	5	0.4812	0.0203	0.607	0.0728	0.5131	0.0183
Z108	2	2.0561	0	2.1156	0.0485	2.0797	0.0248

	3	1.1331	0	1.2286	0.053	1.1678	0.0514
	4	0.6292	0	0.7691	0.0877	0.6762	0.0244
	5	0.4879	0.0116	0.6162	0.0589	0.5197	0.017
Z144	2	1.8075	0	1.8649	0.0354	1.8325	0.024
	3	1.105	0	1.1639	0.0239	1.1238	0.0125
	4	0.6919	0.0083	0.7774	0.0459	0.7721	0.0394
	5	0.4128	0	0.5201	0.059	0.4334	0.0114

De forma similar a la sección 5.1, la Tabla 4.7 informa de la media promediada y los valores de la desviación estándar (σ) de la función de aptitud de ECM-CSA. Cada experimento para cada imagen y nt consiste en 35 evaluaciones independientes del ECM-CSA. Dado que se trata de un problema de minimización, se espera que la media sea lo más baja posible. De los tres algoritmos implementados (CSA, DE y HS), CSA es capaz de encontrar los mejores valores umbral que minimizan la función objetivo.

Tabla 4.8. Resultados de calidad de las versiones de ECM aplicadas a imágenes de RM (El número en el nombre de las imágenes indica el valor del plano z de la RM).

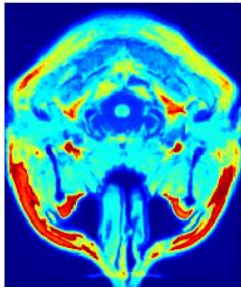
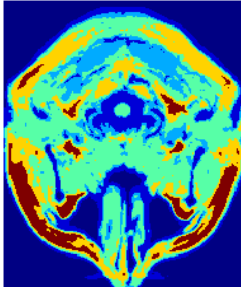
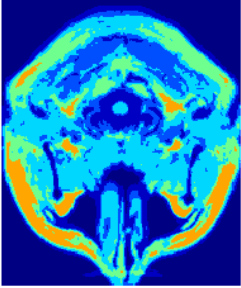
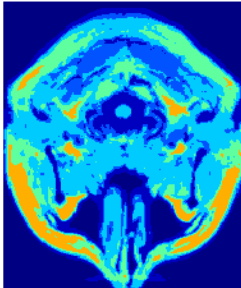
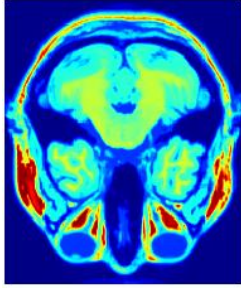
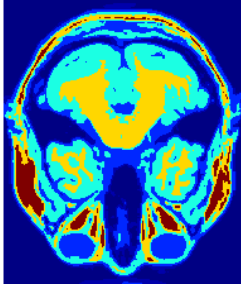
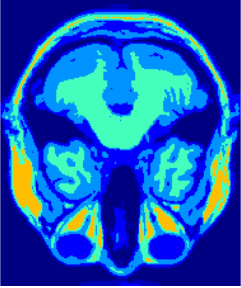
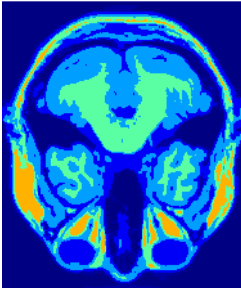
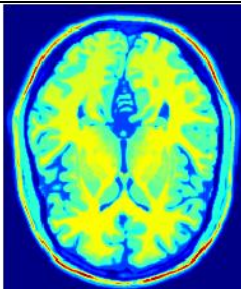
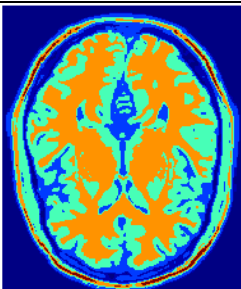
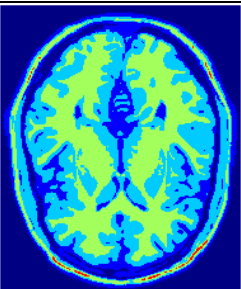
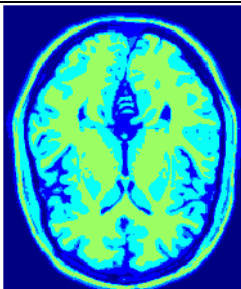
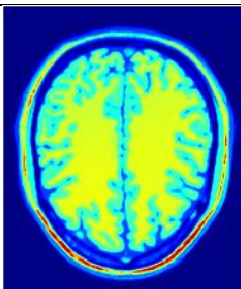
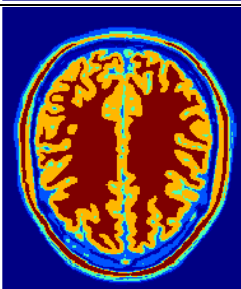
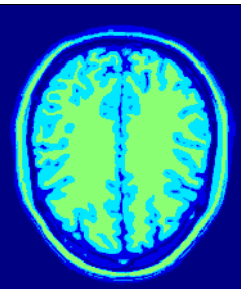
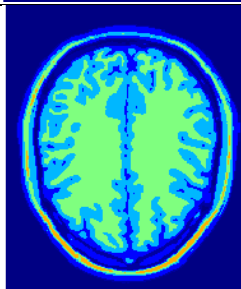
Imagen	nt	CSA			DE			HS		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Z1	2	14.2652	0.4962	0.6076	14.1969	0.4938	0.6046	14.1942	0.4937	0.6046
	3	18.0139	0.6771	0.73	17.8876	0.6728	0.7257	17.9208	0.6737	0.7265
	4	19.8847	0.7377	0.8134	19.7455	0.7322	0.8074	19.7858	0.734	0.8094
	5	21.5104	0.795	0.8484	21.2603	0.7836	0.8364	21.673	0.7897	0.8457
Z2	2	14.3184	0.494	0.6059	14.2475	0.4916	0.603	14.2472	0.4915	0.6029
	3	18.0801	0.6773	0.7326	17.9703	0.6743	0.7291	17.9901	0.6739	0.729
	4	19.9443	0.7366	0.8138	19.7855	0.73	0.8068	19.8458	0.7329	0.8098
	5	21.7553	0.7922	0.8509	21.3264	0.782	0.8389	21.7684	0.7878	0.8474
Z5	2	14.6159	0.4968	0.6114	14.5432	0.4943	0.6084	14.5432	0.4943	0.6084
	3	18.2657	0.6803	0.749	18.1763	0.6758	0.7439	18.1748	0.6769	0.7453
	4	19.9872	0.7305	0.8221	19.8873	0.7262	0.8174	19.886	0.7268	0.8169
	5	21.4894	0.7894	0.8574	21.2333	0.7819	0.8472	21.8387	0.7811	0.8532
Z10	2	16.2405	0.586	0.6642	16.1504	0.583	0.6609	16.1597	0.5831	0.6609
	3	18.6976	0.6701	0.7642	18.5934	0.6656	0.7596	18.6046	0.6668	0.7604
	4	20.2821	0.7085	0.8205	20.1279	0.7023	0.8125	20.1848	0.705	0.8158
	5	21.6823	0.7854	0.8622	21.772	0.7768	0.8548	21.7906	0.7711	0.8557
Z36	2	15.9119	0.6338	0.6807	15.8327	0.6306	0.6773	15.8327	0.6306	0.6773
	3	18.4363	0.7149	0.7844	18.2945	0.7099	0.7798	18.131	0.7065	0.7778
	4	20.175	0.7559	0.8414	20.0269	0.748	0.8332	19.9601	0.753	0.8349
	5	21.2671	0.8209	0.8775	21.0218	0.8082	0.8655	21.163	0.8168	0.8731
Z72	2	15.7358	0.6682	0.7123	15.6544	0.6648	0.7089	15.6575	0.6649	0.7088
	3	19.2135	0.7882	0.8267	19.074	0.7825	0.8211	19.1179	0.7843	0.8226
	4	20.6608	0.85	0.8699	20.4755	0.8416	0.861	20.558	0.8458	0.8656
	5	22.9666	0.8967	0.9042	22.8	0.8849	0.8942	22.9747	0.8947	0.9009
Z108	2	16.1011	0.6862	0.6913	16.0332	0.6831	0.6881	16.021	0.6828	0.6879
	3	19.8096	0.8083	0.79	19.7081	0.8033	0.7844	19.6509	0.8029	0.7848
	4	21.4354	0.8778	0.8538	21.2007	0.8692	0.8441	21.3288	0.8734	0.8496
	5	22.6324	0.8994	0.8823	22.6305	0.8969	0.8746	22.0668	0.8861	0.8717
Z144	2	18.3468	0.7858	0.7616	18.2564	0.7814	0.7575	18.2555	0.7819	0.7578
	3	21.0578	0.835	0.8331	20.9497	0.8305	0.8287	20.953	0.8308	0.829
	4	22.5162	0.8968	0.8675	22.2878	0.8871	0.858	22.4511	0.8636	0.8581
	5	24.2984	0.9132	0.906	24.0175	0.9057	0.8966	24.1757	0.9087	0.9015

Cada imagen segmentada se genera utilizando la Ec. 2.6 y los valores umbral calculados por un algoritmo metaheurístico. En este caso, se utilizan CSA, DE y HS. A partir del análisis cuantitativo, ECM-CSA proporciona evidencia de generar las imágenes segmentadas de más alta calidad de acuerdo a la Tabla 4.8.

En la Tabla 4.9 se presenta una comparación cualitativa de imágenes cerebrales de RM segmentadas. Se seleccionan cuatro imágenes del conjunto de referencia médica para mostrar

visualmente los resultados de los métodos implementados. Para mayor visibilidad, en lugar de mostrar las imágenes en escala de grises, se cambia el mapa de color por el llamado *jet* de Matlab (MathWorks, 2018). Con esta representación, los colores fríos indican baja intensidad mientras que los más cálidos representan valores de alta intensidad. Las imágenes segmentadas con ECM-CSA proporcionan mejores contornos y una representación más precisa.

Tabla 4.9. Resultados cualitativos del algoritmo ECM-CSA aplicados a resonancias magnéticas cerebrales (nt=5).

	Original	CSA	HS	DE
Z1				
Z36				
Z72				
Z108				

4.1.5 Comparaciones

Para evaluar los resultados del enfoque propuesto, se realizan dos comparaciones diferentes. La primera analiza las métricas de desempeño entre CSA, DE y HS cuando se aplican al conjunto de referencia de propósito general. El segundo tiene por objeto evaluar el rendimiento de la metodología propuesta cuando se aplica a las imágenes cerebrales de RM.

4.1.5.1 Comparación de imágenes generales

Para evaluar el rendimiento del algoritmo propuesto, ECM-CSA se compara con dos técnicas de optimización de última generación. Los algoritmos seleccionados para la comparación son Evolución Diferencial (DE) (Rainer Storn & Price, 1997) y Búsqueda de Armonía (HS) (Loganathan, 2001), ambos minimizando la entropía cruzada.

Todos los algoritmos se ejecutan 35 veces sobre cada imagen seleccionada con el número de niveles seleccionado. Las imágenes elegidas para esta prueba son Cameraman, Lena, Livingroom, Blonde, Peppers, Woman Dark Hair, Splash, Baboon, Jet y Sailboat on a lake. En cada prueba se proporcionan los valores de σ (desviación estándar), PSNR, SSIM, FSIM y la media de la función objetivo.

Los resultados se presentan en cuatro tablas, la Tabla 4.4 muestra σ y los valores medios. Las Tablas 4.5, 4.6 y 4.7 presentan métricas de calidad calculadas con el mejor umbral encontrado en las imágenes con umbral. Debe notarse que todos los resultados en la Tabla 4.4, Tabla 4.5, Tabla 4.6 y Tabla 4.7 son promediados a partir de 35 ejecuciones independientes. La Tabla 4.3 lista los mejores valores de umbral encontrados por cada algoritmo.

En este trabajo, la prueba de clasificación de Wilcoxon (sección 2.6.5) se realiza con 35 muestras independientes que consideran un nivel de significancia del 5% sobre el mejor valor de aptitud (entropía cruzada) correspondiente a los cinco umbrales. La Tabla 4.10 presenta los valores p generados por la prueba de Wilcoxon para una comparación por pares del valor de aptitud entre dos grupos formados por CSA vs. DE y CSA vs. HS. Los valores p reportados en la Tabla 4.10 son inferiores a 0,05 (nivel de significancia del 5%), indicando que los valores de aptitud de CSA no ocurrieron por casualidad y fueron evidentemente mejores.

Tabla 4.10. Valores p obtenidos por la prueba de Wilcoxon entre CSA vs. DE y CSA vs. HS para el conjunto de imágenes generales de referencia.

Imagen	nt	p-Valores	
		CSA vs. DE	CSA vs. HS
Cameraman	2	5.32E-06	4.64E-03
	4	5.37E-11	1.04E-11
	8	6.53E-13	6.53E-13
	16	6.50E-13	6.53E-13
	32	6.52E-13	6.50E-13
Lena	2	3.25E-04	1.83E-07
	4	3.47E-11	4.55E-12
	8	6.51E-13	6.52E-13
	16	6.53E-13	6.52E-13
	32	6.49E-13	1.29E-12

Livingroom	2	1.28E-05	6.32E-08
	4	1.27E-10	7.07E-11
	8	6.54E-13	6.54E-13
	16	6.52E-13	6.52E-13
	32	6.48E-13	6.46E-13
Blonde	2	1.81E-05	4.99E-01
	4	1.13E-10	4.75E-12
	8	6.52E-13	6.53E-13
	16	6.51E-13	6.52E-13
	32	6.49E-13	6.47E-13
Peppers	2	2.20E-06	8.06E-06
	4	4.93E-09	2.33E-11
	8	6.53E-13	6.53E-13
	16	6.48E-13	6.52E-13
	32	6.52E-13	6.51E-13
Woman Dark Hair	2	1.31E-05	1.21E-03
	4	1.13E-11	6.54E-13
	8	6.54E-13	6.52E-13
	16	6.53E-13	6.52E-13
	32	6.52E-13	7.66E-13
Splash	2	1.90E-05	4.81E-05
	4	2.42E-11	1.86E-10
	8	6.52E-13	6.54E-13
	16	6.52E-13	6.52E-13
	32	6.52E-13	6.52E-13
Baboon	2	1.18E-05	7.83E-06
	4	2.17E-10	1.53E-12
	8	6.51E-13	6.54E-13
	16	6.49E-13	6.51E-13
	32	6.48E-13	6.49E-13
Jet	2	7.26E-03	5.72E-05
	4	1.01E-09	3.11E-12
	8	6.53E-13	6.49E-13
	16	6.52E-13	6.49E-13
	32	6.42E-13	6.70E-13
Sailboat on lake	2	4.54E-04	4.90E-03
	4	6.79E-11	1.90E-11
	8	6.53E-13	6.52E-13
	16	6.53E-13	6.53E-13
	32	6.47E-13	6.51E-13

4.1.5.2 Comparación de Resonancias Magnéticas

Al igual que en la sección 4.1.5.1 el rendimiento del algoritmo ECM-CSA propuesto se compara con el DE y el HS. En este caso, se selecciona el conjunto de referencia de imágenes médicas. El objetivo de este experimento es determinar si ECM-CSA proporciona o no imágenes cerebrales de RM segmentadas de alta calidad. Puesto que las comparaciones anteriores proporcionaron pruebas que indicaban que ECM-CSA funciona bien en imágenes generales, este experimento se centra en analizar el rendimiento de este tipo de imágenes.

Los resultados se presentan en tres tablas, la Tabla 4.8 muestra la desviación estándar σ y la media. La Tabla 4.9 presenta métricas de calidad calculadas con el mejor umbral encontrado en las imágenes y la Tabla 4.10 proporciona una comparación cualitativa de las imágenes segmentadas. En la Tabla 4.8, la consistencia de los resultados de la CSA es mucho mejor que la σ reportada para DE y HS.

Dado que los resultados proceden de experimentos similares, debe evaluarse la significancia de los datos. Para este propósito, la prueba de rango de Wilcoxon (sección 2.6.5) se realiza con 35 muestras independientes considerando un nivel de significación del 5% considerando el valor de la función objetivo entre dos grupos formados por CSA vs. DE y CSA

vs. HS cuando se aplican en RM cerebrales. Los resultados reportados por la Tabla 4.11 indican que los valores generados por CSA son significativamente distintos a los producidos por DE y HS.

Tabla 4.11.. Valores p obtenidos por la prueba de Wilcoxon entre CSA vs. DE y CSA vs. HS para el conjunto de RM cerebrales.

Imagen	nt	p-Valores	
		CSA vs. DE	CSA vs. HS
Z1	2	6,82E-13	2,35E-12
	3	9,19E-13	3,09E-09
	4	7,11E-13	1,22E-08
	5	1,29E-12	2,35E-05
Z2	2	4,27E-13	4,99E-12
	3	1,18E-12	8,87E-09
	4	1,14E-12	9,24E-09
	5	1,41E-12	8,49E-04
Z5	2	4,22E-13	7,11E-13
	3	1,29E-12	1,01E-09
	4	7,74E-13	9,64E-08
	5	8,82E-13	1,69E-06
Z10	2	5,01E-13	1,14E-12
	3	9,96E-13	5,24E-09
	4	6,53E-13	2,68E-05
	5	6,90E-12	7,67E-03
Z36	2	4,91E-13	1,97E-11
	3	3,00E-12	5,54E-08
	4	1,19E-12	7,17E-04
	5	4,38E-12	1,14E-04
Z72	2	5,49E-13	1,84E-10
	3	1,41E-12	9,97E-08
	4	7,13E-13	1,15E-05
	5	2,55E-12	1,39E-02
Z108	2	6,63E-13	2,81E-12
	3	7,11E-13	1,68E-09
	4	6,53E-13	2,77E-06
	5	2,77E-12	5,53E-04
Z144	2	5,30E-13	4,32E-11
	3	7,76E-13	1,45E-08
	4	7,48E-12	8,70E-10
	5	1,41E-12	2,98E-02

4.1.6 Discusión

Los datos experimentales obtenidos sugieren que la CSA tiene propiedades de búsqueda interesantes cuando se aplica al problema ECM a pesar de sus operadores simples. Una característica importante de la implementación de ECM-CSA es que CSA sólo requiere que se ajusten dos parámetros además del número máximo de iteraciones y el tamaño de la población. ECM-CSA aprovecha las ventajas de varios mecanismos probados que se utilizan comúnmente en algoritmos metaheurísticos. En primer lugar, conviene destacar que la naturaleza poblacional de CSA ayuda a explorar el espacio de búsqueda. Además, ECM trabaja con la información del histograma de la imagen que suele presentar configuraciones multimodales y subóptimos locales. Bajo tal escenario, los algoritmos de búsqueda de una única partícula como HS podrían tener dificultades para escapar del probable estancamiento, lo que llevaría a una convergencia prematura. Otros enfoques basados en población implican el uso de ecuaciones de actualización costosas que pueden ser competitivamente reemplazadas por la ecuación de actualización única de CSA con una generación de posición aleatoria adicional. Dado que la explotación y exploración de la CSA está regulada por la probabilidad de conciencia AP , se convierte en la

parte más sensible del método. Una selección incorrecta del valor AP puede conducir a resultados de segmentación por debajo de lo óptimo. Hay que tener en cuenta que la configuración experimental para los algoritmos basados en la población llamados CSA y DE se establecen en 20. Los resultados señalan que el ECM-CSA propuesto funciona con una población relativamente pequeña.

El enfoque propuesto resulta ser una alternativa interesante a las técnicas tradicionales de umbral de imagen. Además, el ECM-CSA es adecuado para mejorar las imágenes cerebrales de RM tal y como demuestran los resultados. Dado que el procesamiento de imágenes médicas es crucial para el diagnóstico de muchas enfermedades, estos temas han ganado atención en los últimos años. Los resultados competitivos de ECM-CSA podrían animar a los investigadores a aplicar esta técnica a problemáticas similares.

4.2 Imágenes térmicas.

La comunidad de procesamiento de imágenes ha centrado la mayor parte de su atención en el desarrollo de sistemas de visión automática considerando el espectro de luz visible. Sin embargo, existen algunas desventajas a la hora de utilizar este tipo de imágenes en determinadas aplicaciones. La visibilidad de la escena y los colores dependen de una fuente de luz. Con cada cambio de la fuente de luz, la imagen recolectada difiere de la escena original, lo que lleva a escenarios computacionales desafiantes donde el procesamiento de la imagen puede llegar a ser arduo. En el peor de los casos, la ausencia de una fuente de luz hace imposible capturar una imagen. Una gama más amplia del espectro electromagnético puede ser explotada para superar este problema. Concretamente, las imágenes infrarrojas se procesan con las técnicas clásicas de visión por computador que proporcionan buenos resultados en aplicaciones de muchos campos como la agricultura (Vadivambal & Jayas, 2011), la inspección de edificios (Al-Kassir, Fernandez, Tinaut, & Castro, 2005), la detección y el seguimiento de peatones (Leykin & Hammoud, 2010), la validación de sistemas electrónicos (Pitarma, Crisóstomo, & Jorge, 2016) o la asistencia sanitaria (Ring, Ammer, & Ring, 2012) entre otros muchos.

Todos los objetos con una temperatura superior a cero absoluto emiten radiación infrarroja, de suerte que dentro del espectro correspondiente al infrarrojo se encuentra la radiación térmica. Esta radiación térmica puede capturarse con sensores especializados o detectores térmicos. Los objetos emiten radiación en la longitud de onda larga y en la longitud de onda media del espectro infrarrojo (Gade & Moeslund, 2014). Esta radiación varía en la longitud de onda dominante y la intensidad de acuerdo con la temperatura. Las cámaras térmicas toman ventaja de esta propiedad generando imágenes incluso con la falta de una fuente de iluminación (Vollmer, Möllmann, & Wiley InterScience (Online service), 2010). Una representación en escala de grises de la imagen térmica codifica las altas temperaturas como

píxeles brillantes, mientras que los objetos con baja temperatura se representan como píxeles oscuros.

Las imágenes infrarrojas (FLIR, *Forward-looking infrared*) muestran objetos brillantes donde la radiación térmica es mayor. Desde el punto de vista computacional, las imágenes FLIR son idénticas a las imágenes en escala de grises. Sin embargo, las FLIR presentan desafíos adicionales, ya que la resolución de los sensores de captura es generalmente menor que la de las cámaras comunes. Además, las imágenes FLIR típicamente presentan bordes borrosos a medida que el calor irradia. Los contornos difusos dificultan tareas simples como la segmentación, que como se ha indicado previamente se trata de una técnica de pretratamiento bastante común.

Como se indica en la sección 2.5, recopilando lo allí expresado con vistas a su aplicación en lo que respecta a las imágenes FLIR, el objetivo de la segmentación de imágenes es dividir la imagen en clases homogéneas. Cada clase contiene propiedades compartidas como la intensidad o la textura. El caso más sencillo de segmentación de la imagen es la umbralización de imágenes UI, donde se analizan los valores de intensidad del histograma de la imagen para determinar los límites de cada clase. En el umbral de dos niveles, se calcula un valor de umbral para dividir el histograma en dos clases. Esta técnica fue desarrollada para extraer un objeto de su fondo. Para un enfoque más complejo, la umbralización multinivel UM puede identificar un número finito de clases de la imagen. Los métodos de segmentación basados en el umbral pueden dividirse en paramétricos y no paramétricos (Akay & Bahriye, 2013). Los enfoques paramétricos son parámetros estimados de una función de densidad de probabilidad para describir cada clase, pero este enfoque es computacionalmente costoso. Por el contrario, los enfoques no paramétricos utilizan criterios como la varianza entre clases, la entropía y la tasa de error. Estos criterios están optimizados para encontrar el valor umbral óptimo, proporcionando métodos robustos y precisos (Sezgin & Sankur, 2004).

Para reducir el tiempo de cálculo requerido por las técnicas no paramétricas, la umbralización multinivel se calcula con algoritmos evolutivos (EAs). Los EAs comprende algoritmos basados en la interacción de conductas colectivas; usualmente dependiendo de una población de partículas intercambiando información para alcanzar la óptima global. Estos mecanismos proporcionan excelentes propiedades como una búsqueda sin gradientes y pueden evitar el estancamiento en soluciones locales subóptimas.

La metodología analizada en este capítulo trabaja a través de la utilización de varias técnicas de computación evolutiva destinadas a optimizar un criterio particular. Cada EA está diseñado para trabajar con un problema particular bajo ciertas circunstancias. Por lo tanto, ningún algoritmo puede resolver todos los problemas de forma competitiva (Wolpert &

Macready, 1997). Debido a este hecho, una comparación de rendimiento apropiada de cada EA aplicado al umbralización de imágenes FLIR resulta relevante.

4.2.1 Descripción de algoritmos utilizados

Este trabajo está dedicado a analizar el rendimiento del UM por EAs aplicados a las imágenes FLIR. Para ello se seleccionan dos criterios de segmentación: Otsu (1979) y Kapur et al. (1985). Estas metodologías se utilizan como funciones objetivo en los EAs seleccionados. La comparación presta especial atención a los enfoques clásicos como la colonia artificial de abejas (ABC), evolución diferencial (DE), búsqueda de armonías (HS) y el recientemente propuesto algoritmo de polinización de flores (FPA).

4.2.1.1 Colonia artificial de abejas (ABC)

El algoritmo de Colonia Artificial de Abejas, *Artificial Bee Colony* (ABC) propuesto por Karaboga et al. (2007), se ha convertido en una de las técnicas de cómputo evolutivo más usadas por la comunidad. ABC toma su inspiración del comportamiento inteligente que muestran las colonias de abejas durante el forraje. En esta técnica se utiliza una población $\mathbf{L}^k (\{\mathbf{l}_1^k, \mathbf{l}_2^k, \dots, \mathbf{l}_N^k\})$ de N localizaciones de fuentes de comida (soluciones). La población evoluciona de un punto inicial ($k=0$) hasta un número total de generaciones ($k=gen$). Cada localización codifica las variables de diseño del problema a optimizar como un vector d -dimensional $\{l_{i,1}^k, l_{i,2}^k, \dots, l_{i,d}^k\}$. La población \mathbf{L}^k se inicializa aleatoriamente para después evaluar mediante la función objetivo cada una de las localizaciones. El proceso de optimización genera nuevas localizaciones siguiendo los operadores del algoritmo de acuerdo con el tipo de abeja. El primer operador busca una fuente de alimento \mathbf{t}_i en las cercanías de la localización \mathbf{l}_i^k como:

$$\mathbf{t}_i = \mathbf{l}_i^k + \phi (\mathbf{l}_i^k - \mathbf{l}_r^k), \quad i, r \in (1, 2, \dots, N) \quad (4.4)$$

donde \mathbf{l}_r^k se selecciona aleatoriamente ($r \neq i$) y un factor de escalamiento aleatorio ϕ es obtenido de una distribución uniforme entre $[-1, 1]$. Después de generar \mathbf{t}_i , esa nueva solución se evalúa para determinar su valor de aptitud $\text{fit}(\mathbf{l}_i^k)$. En este caso, el valor de aptitud de un problema de minimización se expresa como:

$$\text{fit}(\mathbf{l}_i^k) = \begin{cases} \frac{1}{1 + f(\mathbf{l}_i^k)} & \text{si } f(\mathbf{l}_i^k) \geq 0 \\ \frac{1}{1 + \text{abs}(f(\mathbf{l}_i^k))} & \text{si } f(\mathbf{l}_i^k) < 0 \end{cases} \quad (4.5)$$

donde la función objetivo a minimizar es $f(\cdot)$. Después de calcular los valores de aptitud, el operador de selección codicioso determina si $\text{fit}(\mathbf{t}_i^k)$ es mejor que $\text{fit}(\mathbf{I}_i^k)$. En caso de ser así, la localización \mathbf{I}_i^k es descartada y reemplazada por la nueva solución \mathbf{t}_i^k .

4.2.1.2 Evolución diferencial (DE)

Storn y Price (1997) desarrollaron un algoritmo metaheurístico llamado Evolución Diferencial en inglés *Differential Evolution* (DE). En esta técnica se inicializa una población donde cada elemento contiene vectores con soluciones factibles. Cada individuo se define como:

$$\mathbf{x}_i^k = (x_{1,i}^k, x_{2,i}^k, \dots, x_{N,i}^k), \quad i = 1, 2, \dots, N \quad (4.6)$$

donde \mathbf{x}_i^k es el vector i -ésimo en la generación k . El algoritmo involucra tres pasos principales: mutación, cruce y selección. El proceso de mutación genera un vector \mathbf{x}_i^k en la generación k . Para ello, tres vectores son elegidos aleatoriamente de la población y reciben el nombre de \mathbf{x}_p , \mathbf{x}_q y \mathbf{x}_r . El vector \mathbf{x}_i^k es mutado usando la siguiente ecuación que genera un nuevo vector:

$$\mathbf{v}_i^{k+1} = \mathbf{x}_p^k + F(\mathbf{x}_q^k - \mathbf{x}_r^k) \quad (4.7)$$

donde $F \in [0,1]$ es un parámetro denominado peso diferencial. El segundo operador de cruce es aplicado a la población de acuerdo con la tasa de cruce $C_r \in [0,1]$. Este cruce se inicia con un número aleatorio generado a partir de una distribución uniforme $r_i \in [0,1]$ y el j -ésimo componente de \mathbf{v}_i se manipula de la siguiente forma:

$$\mathbf{u}_{j,i}^{k+1} = \begin{cases} \mathbf{v}_{j,i} & \text{si } r_i \leq C_r \\ \mathbf{x}_{j,i}^k & \text{en otro caso} \end{cases}, \quad j = 1, 2, \dots, d, \quad i = 1, 2, \dots, n \quad (4.8)$$

Finalmente, la etapa de selección se realiza mediante la comparación del valor de aptitud del vector candidato contra el vector original:

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{u}_i^{t+1} & \text{si } f(\mathbf{u}_i^{t+1}) \leq f(\mathbf{x}_i^t) \\ \mathbf{x}_i^t & \text{en otro caso} \end{cases} \quad (4.9)$$

4.2.1.3 Búsqueda de armonías (HS)

El algoritmo de búsqueda de armonías, *Harmony Search* en inglés (HS) fue presentado por Geem et al. (2001). Este algoritmo metaheurístico fue inspirado por el proceso de

improvisación realizado durante la interpretación de Jazz. Para ello se genera una población denominada memoria de armonías $\mathbf{HM}^k \left(\{ \mathbf{H}_1^k, \mathbf{H}_2^k, \dots, \mathbf{H}_N^k \} \right)$ con N partículas (armonías). El proceso de optimización evoluciona la memoria \mathbf{HM}^k desde un punto inicial ($k=0$) hasta un total de iteraciones ($k=gen$). Cada armonía \mathbf{H}_i^k representa un vector de variables de decisión d -dimensional del problema a optimizar $\{ H_{i,1}^k, H_{i,2}^k, \dots, H_{i,N}^k \}$. HS funciona generando nuevas armonías considerando las presentes en la memoria \mathbf{HM}^k . Inicialmente la memoria se inicializa aleatoriamente. Después se genera una nueva solución candidata usando los elementos de la memoria con un ajuste de tono o una re-inicialización aleatoria. A la generación de una nueva armonía se le llama improvisación y ésta depende de la tasa de consideración de la memoria de armónicos (HMCR). Si tal parámetro se establece muy pequeño, sólo unos cuantos miembros de la memoria serán considerados para la fase de exploración. En este paso, el valor de la primera variable de decisión de la nueva armonía $H_{new,1}$ se selecciona aleatoriamente de entre los valores de cualquier armonía existente en la memoria para la misma variable de decisión. Este proceso se repite para todas las variables de decisión

$$H_{new} = \begin{cases} H_j \in \{x_{1,j}, x_{2,j}, \dots, x_{HMS,j}\} & \text{con probabilidad HMCR} \\ \text{generada aleatoriamente} & \text{con probabilidad (1-HMCR)} \end{cases} \quad (4.10)$$

Todos los valores $H_{new,1}$ son perturbados con una operación llamada ajuste de tono. En esta fase, la tasa de ajuste de tono (PAR) es la frecuencia con la que se ajustará, además el factor de ancho de banda (BW) controla el grado con el que se aplica la perturbación

$$H_{new} = \begin{cases} H_{new} \pm \text{rand}(0,1) \cdot BW & \text{con probabilidad PAR} \\ H_{new} & \text{con probabilidad (1-PAR)} \end{cases} \quad (4.11)$$

Finalmente, la memoria \mathbf{HM}^k se actualiza si la nueva armonía es mejor que la peor de la memoria.

4.2.1.4 Algoritmo de polinización de flores (FPA)

El algoritmo de polinización de flores, *Flower Pollintation Algorithm* (FPA), propuesto por Yang (2012), es un algoritmo metaheurístico basado en el proceso de polinización de las flores. En el FPA, las soluciones representan un conjunto de flores que respetan las reglas biológicas del proceso de polinización. Desde el punto de vista computacional se usa una población $\mathbf{F}^k \left(\{ \mathbf{f}_1^k, \mathbf{f}_2^k, \dots, \mathbf{f}_N^k \} \right)$ de N soluciones (posiciones de flores). Tal población evoluciona desde un punto inicial ($k=0$) hasta un número total de generaciones ($k=gen$). Cada flor $\mathbf{f}_i^k (i \in [1, \dots, N])$ representa un vector d -dimensional $\{ \mathbf{f}_{i,1}^k, \mathbf{f}_{i,2}^k, \dots, \mathbf{f}_{i,N}^k \}$ donde cada dimensión

corresponde a una variable de decisión del problema a resolver. En cada generación se produce una nueva población considerando dos operadores: polinización local y polinización global. Para elegir entre los dos operadores se utiliza un factor de polinización global p . Para cada flor \mathbf{f}_i^k se generará un número r_p con una distribución uniforme en el rango $[0,1]$. Si r_p es mayor que p , el operador de polinización local se aplica a \mathbf{f}_i^k ; en caso contrario se aplicará el operador de polinización global.

En el operador de polinización global, la posición original \mathbf{f}_i^k se desplaza a una nueva posición \mathbf{f}_i^{k+1} de acuerdo con el siguiente modelo:

$$\mathbf{f}_i^{k+1} = \mathbf{f}_i^k + s_i \cdot (\mathbf{f}_i^k - \mathbf{g}) \quad (4.12)$$

donde \mathbf{g} es la mejor posición encontrada hasta el momento. Además s_i controla la longitud del desplazamiento. Para ello se usa una distribución de Lévy simétrica que genera los valores s_i siguiendo el algoritmo de Mantegna (Mantegna, 1994):

$$s_i = \frac{\mathbf{u}}{|\mathbf{v}|^{1/\beta}} \quad (4.13)$$

donde $\mathbf{u}(\{u_1, \dots, u_d\})$ y $\mathbf{v}(\{v_1, \dots, v_d\})$ son vectores d -dimensionales y $\beta = 3/2$. Cada elemento de \mathbf{u} y \mathbf{v} se calcula considerando la siguiente distribución normal:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2) \quad (4.14)$$

$$\sigma_u = \left(\frac{\Gamma(1+\beta) \cdot \sin(\pi \cdot \beta / 2)}{\Gamma((1+\beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}} \right)^{1/\beta}, \quad \sigma_v = 1 \quad (4.15)$$

donde $\Gamma(\cdot)$ indica una distribución Gamma.

En el caso del operador de polinización local, la posición actual \mathbf{f}_i^k se perturba dando como resultado \mathbf{f}_i^{k+1} . La perturbación se define como:

$$\mathbf{f}_i^{k+1} = \mathbf{f}_i^k + \varepsilon \cdot (\mathbf{f}_j^k - \mathbf{f}_h^k), \quad i, j, h \in (1, 2, \dots, N) \quad (4.16)$$


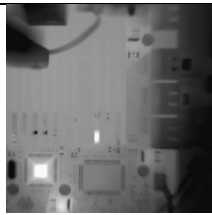
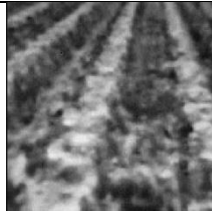
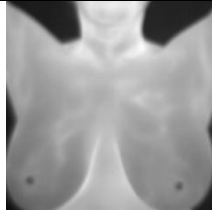

donde \mathbf{f}_j^k y \mathbf{f}_h^k son dos flores elegidas aleatoriamente satisfaciendo la condición $i \neq j \neq h$. También se utiliza un factor de escalamiento ε extraído de una distribución normal entre $[-1,1]$.

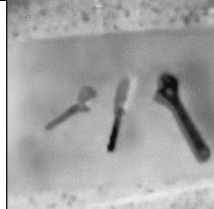

A continuación se analiza el rendimiento de algunos algoritmos metaheurísticos aplicados al problema de la umbralización de imágenes, específicamente a la segmentación de imágenes FLIR. Los siguientes experimentos están diseñados para maximizar la función de aptitud del método Otsu o de Kapur Ec. 2.14 y Ec. 2.22 respectivamente.

4.2.2 Montaje experimental

Puesto que el objetivo es analizar el comportamiento de MTH con EAs en imágenes FLIR, se prepara un grupo diverso de prueba con siete imágenes procedentes de diversas escenas y campos del estudio, las cuales se encuentran en la tabla 4.12. Todas las imágenes son JPEG de 480x480 y sus propiedades específicas se describen en la Tabla 4.12. Los experimentos se realizaron usando Matlab 8.3 (MatWorks, 2018) en una CPU i5-4210 @2.3Ghz con 6GB de RAM.

Tabla 4.12. Imágenes térmicas de prueba.

Nombre	Dispositivo de captura	Origen	Imagen
Cars	FLIR One	Capturado por el autor.	
Circuit	FLIR SC660	Multispectral Image Database (Zukal, Mekyska, Cika, & Smekal, 2013).	
Crop	FLIR One	Capturado por el autor.	
Medical	FLIR SC-620	Database for Mastology Research (Silva, L. F.; Saade, D. C. M.; Sequeiros, G. O.; Silva, A. C.; Paiva, A. C.; Bravo, R. S.; Conci, 2014)	
Office	FLIR One	Capturado por el autor.	

Tools	FLIR One	Capturado por el autor.	
Pedestrian	Raytheon 300D	OTCBVS Benchmark Dataset 01(Keck, 2005)	

Para cada evaluación del algoritmo, se establece un criterio de parada de 3000 generaciones. La calidad de la segmentación se evalúa utilizando métodos estadísticos estándar y métricas de calidad de la señal. Cada algoritmo metaheurístico requiere de diversos parámetros para funcionar apropiadamente. Para este experimento se utilizaron las configuraciones descritas en la tabla 4.13.

Tabla 4.13. Parámetros de implementación.

Algoritmo	Parámetros
ABC	El algoritmo se ha implementado utilizando las directrices proporcionadas por su referencia (Dervis Karaboga & Basturk, 2007)
DE	La variante implementada es DE/rand/bin donde $cr = 0.5$ y el peso diferencial = 0.2 (Rainer Storn & Price, 1997)
HS	El algoritmo se implementó utilizando las directrices proporcionadas por su referencia (Geem, Kim, & Loganathan, 2001) con $HMCR = 0.95$ y $PAR = 0.3$.
FPA	El algoritmo sigue siendo el descrito en su propuesta (X. S. Yang, 2012)

Los valores reportados en cada experimento de la Tabla 4.14 a la Tabla 4.19 son los resultados promediados de 35 evaluaciones de cada función objetivo con su respectivo algoritmo en cada imagen y número de umbrales nt. La Tabla 4.14 y la Tabla 4.17 reportan datos estadísticos del método Otsu (1979) y Kapur et al. (1985) respectivamente. Las tablas 4.15 y 4.16 están divididas para facilitar su lectura. Lo mismo ocurre con los cuadros 4.18 y 4.19. El mejor valor de cada experimento aparece resaltado en negrita.

4.2.3 Resultados usando la varianza entre clases

Dado que el método Otsu es un proceso de maximización, se espera que la función objetivo sea lo más alta posible. La Tabla 4.14 muestra el valor medio y la desviación estándar (σ) de la función objetivo de Otsu para todos los algoritmos analizados. Se observa cómo FPA supera a ABC, DE y HS en ambas métricas.

Aunque FPA supera a otros algoritmos maximizando la función objetivo y proporcionando resultados consistentes, los indicadores de calidad de la imagen segmentada no reflejan este comportamiento. La Tabla 4.15 y la Tabla 4.16 muestran las métricas de calidad FSIM, SSIM y PSNR. Estas métricas muestran un comportamiento disperso; el HS es el

algoritmo con la mayor cantidad de mejores valores experimentales y es seguido de cerca por el FPA. El EA que acumuló menos valores experimentales mejores es DE.

Tabla 4.14. Resultados estadísticos de ABC, DE, HS y FPA utilizando el método Otsu como función objetivo.

Imagen	nt	ABC		DE		HS		FPA	
		Media	σ	Media	σ	Media	σ	Media	σ
Cars	2	1190.9202	0.3017	1191.2043	0.0707	1191.2496	0.0169	1191.2578	0.0049
	3	1614.8457	0.5955	1614.4202	0.9947	1615.4343	0.3444	1615.8415	0.0339
	4	1714.3754	0.9779	1713.7907	3.5836	1715.9867	1.0195	1717.1537	0.0666
	5	1766.6552	1.3444	1765.8426	2.6367	1768.04	1.8868	1771.1984	0.12
Circuit	2	1968.1636	0.158	1968.4058	0.1277	1968.4245	0.0686	1968.4741	0.0053
	3	2258.5519	0.4958	2258.2495	0.4141	2258.8803	0.2944	2259.117	0.033
	4	2327.1105	6.1012	2343.6385	0.9983	2345.1314	0.5103	2345.7375	0.0505
	5	2386.8003	5.6086	2401.27	1.7315	2404.0981	0.4784	2405.008	0.0945
Crop	2	1538.5842	0.1754	1538.5873	0.2875	1538.6882	0.042	1538.7057	0.0047
	3	1841.6442	0.4689	1841.4914	0.7492	1842.1634	0.3402	1842.4936	0.0276
	4	1961.134	0.7381	1959.8387	1.5208	1962.0959	0.6112	1962.8465	0.0572
	5	2018.9002	1.3835	2017.1197	2.7514	2020.7496	1.2512	2022.3693	0.096
Medical	2	1588.5637	0.0943	1588.7291	0.0383	1588.747	0.0278	1588.7605	0.0084
	3	2073.2428	0.8432	2073.3506	0.7975	2073.9934	0.2724	2074.3334	0.0294
	4	2197.6308	2.2985	2199.1006	1.8829	2200.2549	1.4825	2202.0307	0.0942
	5	2247.3319	1.934	2247.4678	1.8325	2247.792	2.9289	2251.6854	0.0826
Office	2	1392.2126	0.3049	1392.6247	0.1388	1392.6854	0.0681	1392.7315	0.0028
	3	1673.6714	3.499	1679.5073	0.6906	1680.1208	0.2712	1680.4159	0.0302
	4	1859.8373	1.1902	1859.4691	1.937	1861.6973	1.0111	1862.8328	0.085
	5	1933.3062	1.2189	1932.5815	3.5319	1936.4498	0.9864	1938.071	0.0878
Tool	2	674.3096	0.2216	674.5791	0.1487	674.5937	0.2727	674.7168	0.0071
	3	1005.9789	0.8258	1006.1544	1.0812	1006.8794	0.2997	1007.1833	0.0328
	4	1102.898	1.4988	1104.1747	1.0409	1104.7636	1.4774	1106.5584	0.0471
	5	1160.218	1.7741	1158.2793	3.2629	1161.629	1.5385	1163.8411	0.1139
Pedestrian	2	111.9835	0.1632	112.1431	0.0789	112.1686	0.0735	112.2064	0.0024
	3	158.7331	0.678	159.6061	0.201	159.7849	0.1075	159.8552	0.0058
	4	188.4953	0.5837	188.5496	0.3837	189.0816	0.1678	189.3136	0.0246
	5	200.6837	0.7315	201.0701	0.6576	202.0394	0.1989	202.3881	0.0285

Tabla 4.15. Resultados de calidad de la imagen segmentada para ABC y DE utilizando el método Otsu.

Imagen	nt	ABC			DE		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Cars	2	10.2954	0.2538	0.6705	9.4639	0.1638	0.6662
	3	13.9399	0.4901	0.7046	14.6155	0.4963	0.7144
	4	15.2883	0.5149	0.7401	15.5772	0.5254	0.7493
	5	16.975	0.5824	0.7495	17.3467	0.5761	0.7662
Circuit	2	13.6374	0.4528	0.737	14.6589	0.4956	0.7367
	3	17.1852	0.5556	0.7701	18.084	0.5839	0.7769
	4	18.8839	0.625	0.7919	20.7318	0.6653	0.8036
	5	20.5025	0.6627	0.8051	20.8779	0.6671	0.8087
Crop	2	12.0396	0.2526	0.5341	11.638	0.2201	0.5291
	3	14.2759	0.3547	0.6136	14.4982	0.3498	0.6286
	4	15.6913	0.4288	0.6673	16.7028	0.4539	0.7026
	5	17.9979	0.5247	0.7306	18.9015	0.5538	0.7669
Medical	2	13.3951	0.7284	0.8128	13.522	0.7489	0.8186
	3	15.3872	0.7514	0.8157	16.2999	0.768	0.8214
	4	16.2494	0.7524	0.8212	17.7156	0.7697	0.8211
	5	18.5999	0.7773	0.8211	19.8676	0.7771	0.8167
Office	2	10.7762	0.2864	0.7355	10.8032	0.2626	0.7332
	3	13.8166	0.4906	0.74	15.5397	0.5631	0.7348
	4	15.9192	0.6302	0.7559	16.49	0.5975	0.7466
	5	19.0038	0.7143	0.7791	20.2068	0.7352	0.7819
Tool	2	9.9161	0.4581	0.6241	9.4845	0.4369	0.609
	3	15.097	0.6379	0.689	16.3156	0.6705	0.6899
	4	17.6627	0.6601	0.7173	18.5422	0.6669	0.7237
	5	20.4009	0.731	0.7827	21.7611	0.7301	0.7953
Pedestrian	2	13.5776	0.2426	0.57	13.2081	0.235	0.5772
	3	14.0902	0.2733	0.5933	14.2089	0.2862	0.6031
	4	18.1878	0.4948	0.6706	17.4529	0.4682	0.6699
	5	21.0671	0.6264	0.7408	21.7515	0.671	0.7406

Tabla 4.16. Resultados de calidad de la imagen segmentada para HS y FPA utilizando el método Otsu.

Image	nt	HS			FPA		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Cars	2	9.5091	0.1663	0.6705	9.4615	0.1648	0.6673
	3	14.6123	0.4951	0.713	14.6126	0.4949	0.7126
	4	15.6444	0.5252	0.7502	15.6022	0.5253	0.7511
	5	17.8162	0.593	0.7631	17.3501	0.5765	0.7667
Circuit	2	14.9888	0.5036	0.7464	15.0464	0.5097	0.7507
	3	18.1943	0.5848	0.7782	18.1592	0.5825	0.7761
	4	20.7454	0.6671	0.8049	20.6473	0.6636	0.8016
	5	20.9371	0.6672	0.8103	20.8335	0.6661	0.8065
Crop	2	11.774	0.2242	0.5403	11.6126	0.2191	0.5266
	3	14.4809	0.3491	0.6259	14.5093	0.3516	0.6296
	4	16.7014	0.4547	0.7024	16.7147	0.4559	0.7034
	5	19.0409	0.5598	0.7695	18.9596	0.5569	0.769
Medical	2	14.3094	0.7585	0.8188	14.4381	0.7582	0.8189
	3	16.6016	0.7645	0.8197	16.3592	0.7688	0.8216
	4	17.7807	0.7685	0.8215	17.7072	0.7704	0.8215
	5	20.1848	0.7818	0.8202	19.7926	0.7773	0.8171
Office	2	10.7984	0.2623	0.7327	10.7272	0.2631	0.7329
	3	15.5333	0.563	0.7339	15.4713	0.5634	0.7347
	4	16.4959	0.5971	0.7463	16.4532	0.5975	0.7452
	5	20.3213	0.7387	0.7836	20.1876	0.7382	0.782
Tool	2	9.5028	0.44	0.6114	9.4829	0.4384	0.6095
	3	16.3974	0.6691	0.6916	16.3915	0.6698	0.6916
	4	18.5818	0.665	0.7253	18.5346	0.668	0.7229
	5	21.7453	0.7307	0.7959	21.7444	0.7307	0.7936
Pedestrian	2	13.2043	0.2339	0.5761	13.2042	0.2341	0.5766
	3	14.2107	0.2864	0.6038	14.2103	0.2863	0.6037
	4	17.4041	0.466	0.6702	17.4016	0.4658	0.6694
	5	21.8061	0.6727	0.7442	21.6591	0.6675	0.7385

4.2.4 Resultados usando la entropía máxima

En la Tabla 4.17 se presentan los datos estadísticos del método Kapur aplicados al EA seleccionado. La función objetivo también maximiza la entropía Kapur de cada imagen para cada experimento. Como resultado, un valor más alto de la media de la función objetivo indica una mejor segmentación. Contrariamente al método Otsu, ABC supera al resto en la mayoría de los experimentos.

Además, las Tablas 4.18 y 4.19 reportan métricas de calidad sin un claro ganador. El algoritmo que supera la mayoría de los experimentos es HS. ABC y DE muestran una cantidad similar de mejores valores y FPA posee el menor número de mejores valores.

Tabla 4.17. Resultados estadísticos de ABC, DE, HS y FPA utilizando la entropía de Kapur como función objetivo.

Imagen	nt	ABC		DE		HS		FPA	
		Media	σ	Media	σ	Media	σ	Media	σ
Cars	2	17.6758	0.0014	17.6628	0.0087	17.6644	0.0149	17.6194	0.0346
	3	22.1336	0.0056	22.0928	0.0171	22.109	0.027	22.025	0.0568
	4	26.1868	0.0827	25.9645	0.074	26.0083	0.0549	25.9566	0.1526
	5	30.0658	0.0442	29.8316	0.1262	29.8338	0.0856	29.7889	0.1707
Circuit	2	17.3273	0.0018	17.293	0.0137	17.3012	0.0183	17.2077	0.0587
	3	21.6443	0.0298	21.5546	0.06	21.5241	0.0475	21.4583	0.1298
	4	25.8571	0.0185	25.7014	0.0641	25.7563	0.0677	25.6115	0.1012
	5	29.639	0.027	29.4349	0.0628	29.5821	0.0446	29.2699	0.2038
Crop	2	17.8285	0.0012	17.8174	0.0068	17.8133	0.0091	17.7597	0.0532
	3	22.2342	0.0036	22.2152	0.0099	22.2112	0.0197	22.1498	0.0468
	4	26.3697	0.0137	26.313	0.0212	26.3055	0.0459	26.2505	0.0729
	5	30.417	0.0697	30.1759	0.0888	30.2176	0.0449	30.1592	0.1782
Medical	2	17.5238	0.0019	17.5033	0.0087	17.4839	0.0304	17.449	0.0478
	3	21.8562	0.0023	21.822	0.0156	21.8314	0.0146	21.763	0.0612
	4	26.2693	0.0063	26.1912	0.0293	26.2363	0.0201	26.0938	0.0811

	5	30.2272	0.021	30.0929	0.0569	30.159	0.0939	30.0474	0.072
Office	2	17.3833	0.002	17.3661	0.0085	17.3682	0.016	17.2859	0.0652
	3	21.785	0.0038	21.7344	0.0176	21.7633	0.0194	21.65	0.0687
	4	25.9259	0.09	25.6474	0.1458	25.6545	0.1159	25.533	0.1824
	5	29.7348	0.0513	29.4132	0.1638	29.4566	0.126	29.3374	0.1791
Tool	2	17.393	0.0033	17.3808	0.0075	17.36	0.0464	17.3274	0.0447
	3	21.7232	0.0095	21.6974	0.0134	21.6217	0.0766	21.6505	0.0592
	4	25.828	0.0278	25.7369	0.027	25.7745	0.0338	25.6829	0.0759
	5	29.8829	0.0537	29.6592	0.0945	29.6438	0.1062	29.588	0.1327
Pedestrian	2	16.8579	0.0052	16.8318	0.0138	16.8102	0.0257	16.7485	0.0539
	3	21.4753	0.006	21.3942	0.0315	21.4241	0.0248	21.2526	0.1067
	4	25.6262	0.0172	25.4739	0.0451	25.4842	0.0806	25.3198	0.1225
	5	29.2976	0.0168	29.0905	0.0555	29.1945	0.0738	28.9653	0.1283

Tabla 4.18. Resultados de calidad de la imagen segmentada para ABC y DE utilizando la entropía de Kapur.

Image	nt	ABC			DE		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Cars	2	13.2778	0.4617	0.69	14.1686	0.4736	0.7007
	3	14.6202	0.5192	0.7156	15.1789	0.5082	0.7323
	4	15.9592	0.5863	0.722	17.6729	0.5936	0.7527
	5	17.7815	0.6542	0.7437	18.6284	0.623	0.7658
Circuit	2	14.8876	0.5145	0.7488	14.6412	0.4987	0.7505
	3	15.0393	0.5502	0.7581	15.5329	0.5251	0.7627
	4	17.5823	0.6173	0.7825	17.895	0.5958	0.7859
	5	19.0199	0.6498	0.7942	20.8381	0.6563	0.8144
Crop	2	12.6244	0.2927	0.5547	13.144	0.2785	0.5822
	3	15.8271	0.4413	0.6381	15.3553	0.3836	0.6638
	4	15.7842	0.4608	0.6398	17.9957	0.5012	0.7209
	5	18.2075	0.5553	0.709	20.0144	0.5871	0.7646
Medical	2	11.2497	0.6328	0.8083	14.4628	0.7255	0.7849
	3	14.8435	0.7613	0.822	15.4723	0.7694	0.8175
	4	17.2505	0.7833	0.8285	19.2433	0.7979	0.8458
	5	17.8286	0.791	0.8352	20.7799	0.8104	0.8513
Office	2	12.3584	0.4383	0.7375	12.3439	0.3739	0.7204
	3	15.6265	0.6365	0.7568	17.8563	0.7161	0.762
	4	16.8774	0.6918	0.7681	18.3948	0.7223	0.7677
	5	17.6706	0.7116	0.7756	19.6462	0.7351	0.7774
Tool	2	14.1762	0.6414	0.6905	16.5409	0.6671	0.6814
	3	15.8012	0.6834	0.7086	17.9838	0.699	0.7114
	4	17.614	0.7002	0.7299	18.7458	0.7173	0.7433
	5	16.4082	0.6975	0.734	20.063	0.7224	0.7641
Pedestrian	2	14.7716	0.2837	0.5838	10.1687	0.0149	0.5534
	3	17.316	0.441	0.6171	17.5064	0.4588	0.6222
	4	18.3683	0.5001	0.6617	17.5323	0.4609	0.6271
	5	18.133	0.4842	0.6761	19.9934	0.5994	0.7582

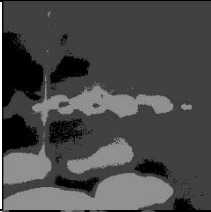







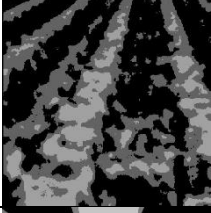
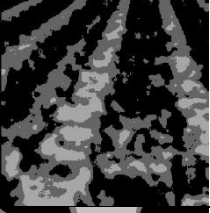
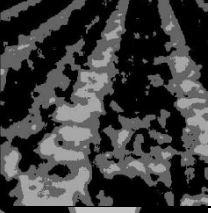
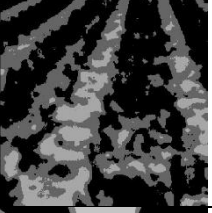




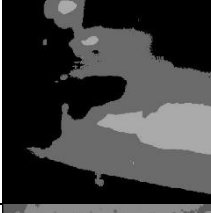
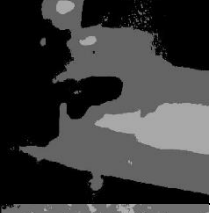




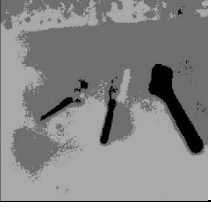

Tabla 4.19. Resultados de calidad de la imagen segmentada para HS y FPA utilizando la entropía de Kapur.

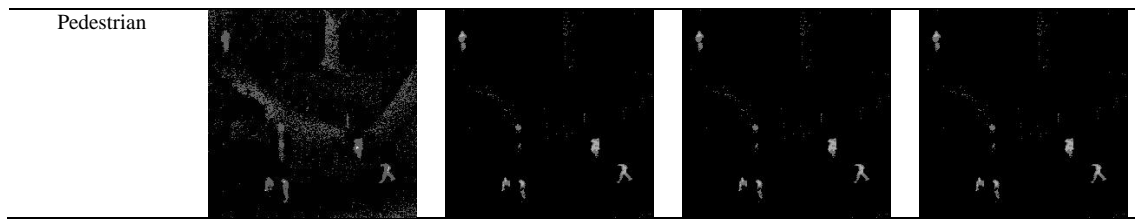
Imagen	nt	HS			FPA		
		PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Cars	2	14.1711	0.4738	0.7008	14.1686	0.4736	0.7007
	3	15.1759	0.508	0.7322	15.1772	0.508	0.7323
	4	16.3196	0.5408	0.7385	15.6756	0.5284	0.7321
	5	18.0457	0.6088	0.7602	17.527	0.5848	0.7559
Circuit	2	14.6538	0.4981	0.7501	14.6429	0.4988	0.7505
	3	16.2929	0.5359	0.7651	14.6147	0.4998	0.7562
	4	17.8598	0.5978	0.785	17.9018	0.5976	0.7854
	5	20.6348	0.6642	0.8163	20.6868	0.6635	0.8154
Crop	2	13.1881	0.2808	0.5819	13.144	0.2785	0.5822
	3	15.801	0.4031	0.6625	15.3606	0.3839	0.6638
	4	18.533	0.5258	0.7276	17.3676	0.4727	0.7061
	5	18.0291	0.5098	0.7422	18.583	0.5293	0.7338
Medical	2	14.4628	0.7255	0.7849	14.4628	0.7255	0.7849
	3	15.3062	0.7702	0.8183	15.2861	0.7701	0.8184
	4	19.2404	0.7979	0.8458	19.2474	0.7982	0.8457
	5	20.5062	0.815	0.8518	20.6473	0.8114	0.8518
Office	2	12.3439	0.3739	0.7204	12.3439	0.3739	0.7204
	3	17.872	0.723	0.7638	17.8736	0.7178	0.7624
	4	18.1053	0.7312	0.7678	17.9242	0.7171	0.7652
	5	19.6923	0.7717	0.783	18.9943	0.7316	0.7722

Tool	2	16.523	0.6875	0.6958	16.5409	0.6671	0.6814
	3	18.0815	0.7011	0.7162	17.9719	0.6992	0.7114
	4	18.7427	0.7132	0.7371	18.6826	0.7083	0.7313
	5	19.8386	0.7241	0.7627	19.8999	0.7196	0.7553
Pedestrian	2	10.1686	0.0149	0.5534	10.1687	0.0149	0.5534
	3	17.506	0.4587	0.6222	17.2622	0.444	0.6202
	4	17.5389	0.4609	0.6268	17.5639	0.4618	0.6273
	5	19.8776	0.5932	0.7526	19.925	0.595	0.754

La Tabla 4.20 muestra una comparación cualitativa de los enfoques propuestos con tres valores umbral ($nt=3$) como ejemplo. La diferencia en las imágenes presentadas no es fácil de detectar a simple vista. Los cuatro enfoques presentan resultados similares que sólo pueden ser cuantificados por las métricas correspondientes, tal y como se ha indicado previamente.

Tabla 4.20. Resultados cualitativos utilizando la entropía de Kapur.

	ABC	DE	HS	FPA
Cars				
Circuit				
Crop				
Medical				
Office				
Tool				



4.3 Resumen.

En este capítulo se presenta el análisis del desempeño de EAs aplicados a imágenes especiales. En la primera parte del capítulo se analiza el desempeño de la entropía cruzada con algoritmos metaheurísticos para la segmentación de resonancias magnéticas cerebrales. En la segunda parte se presenta una comparación entre algoritmos clásicos usando la varianza entre clases y la entropía máxima para la segmentación de imágenes térmicas. En ambos casos se ha observado que la calidad de las imágenes segmentadas está fuertemente ligada al desempeño del algoritmo evolutivo seleccionado. Es por ello que resulta importante analizar el rendimiento de la umbralización para cada tipo de imágenes en específico. De esta forma, es posible seleccionar el algoritmo que reporte mejores resultados en cada aplicación.

5 Umbralización basada en contexto

Como se ha expuesto reiteradamente a lo largo de la memoria, la umbralización es una herramienta útil y sencilla en segmentación de imágenes, existiendo una amplia diversidad de técnicas con aportaciones continuas. Sin embargo, la mayoría de las investigaciones propuestas se enfocan en realizar la umbralización sobre el histograma de la imagen. Recientemente se introdujo el concepto de la curva de energía como método de umbralización para considerar la información espacial alrededor de los píxeles (Ghosh, Bruzzone, Patra, Bovolo, & Ghosh, 2007; Patra, Gautam, & Singla, 2014). En consecuencia, el objetivo de este capítulo se centra en este aspecto evaluando los resultados derivados de la incorporación de este concepto en algoritmos tradicionales y más recientes.

La umbralización es una técnica de segmentación de imágenes que goza de simplicidad, robustez y precisión (Hammouche et al., 2010; Sezgin & Sankur, 2004). Sin embargo, la umbralización multinivel con métodos clásicos genera problemas dado que tales implementaciones suelen estar asociadas a un elevado coste computacional para encontrar los valores de umbral. La umbralización de imágenes ha sido tratada mediante el uso de algoritmos evolutivos para realizar la búsqueda de los mejores valores de umbral sin incrementar en gran medida el coste computacional. Algunas contribuciones recientes son muestras de la validez de este enfoque. La aplicación de algoritmos metaheurísticos recientemente aplicados al problema de la umbralización son buena muestra de ello (Aziz, Ewees, & Hassanien, 2017; Dehshibi et al., 2017; Hussein, Sahran, & Abdullah, 2016; L. Li et al., 2017; Suresh & Lal, 2017).

A pesar de que los enfoques basados en el histograma son los más utilizados, se han realizado esfuerzos por incorporar información espacial al proceso de la segmentación basada en umbrales. Una técnica relevante que incorporara información espacial es el trabajo de Ghosh

et al. (2007), adaptado posteriormente al algoritmo Cuckoo Search algorithm (Pare, Kumar, Bajaj, & Singh, 2016) y más tarde ampliado mediante el uso de matrices de co-ocurrencia para intensidades de grises (Pare, Bhandari, Kumar, & Singh, 2017). Al ser relativamente nueva la incorporación de la curva de energía al proceso de umbralización aún no existe en la literatura una evaluación exhaustiva del potencial de esta nueva metodología.

Bajo las consideraciones anteriores, este capítulo se basa en los resultados publicados en (Oliva, Hinojosa, Elaziz, & Ortega-Sánchez, 2018) teniendo la siguiente organización: en la sección 5.1 se introduce el concepto de curva de energía. En la sección 5.2 se describen dos algoritmos de naturaleza contextual. La sección 5.3 incluye el análisis de los resultados obtenidos, que se completa con una valoración al respecto en la sección 5.4.

5.1 Curva de energía

La curva de energía es un medio de representación de la información contenida en una imagen al igual que el histograma. La curva de energía tiene propiedades interesantes tales como la incorporación de información espacial o contextual de la imagen y no solo la intensidad de cada píxel, que constituye la diferencia clave con respecto al histograma. En comparación con los histogramas, las curvas de energía son visualmente más suaves, conservando valles y picos. En este caso, la umbralización de imágenes consiste en la búsqueda de los umbrales correspondientes a la mitad de las regiones de valles existentes en la curva de energía. Cada valle existe entre dos modos adyacentes, cada modo caracteriza un objeto de la imagen. Así, el objetivo que se plantea es la selección precisa de valores de umbral para obtener los mejores resultados posibles en la segmentación.

Para este proceso una imagen I es usada definiéndola como la matriz $I = \{I_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ de tamaño $m \times n$ donde I_{ij} representa al valor de intensidad de la imagen I en el píxel (i, j) . El valor máximo de intensidad de gris en la imagen I se denota como L . Para considerar la información contextual espacial es necesario determinar una correlación espacial entre los píxeles que rodean a uno dado en particular. Para ello, se define una vecindad N de orden d en una posición dada (i, j) , $N_{ij}^d = \{(i+u, j+v), (u, v) \in N^d\}$. El valor d determina la configuración de vecindad a utilizar (Ghosh et al., 2007). En esta investigación se considera una vecindad de segundo orden N^2 . El sistema puede definirse en términos espaciales como $(u, v) \in \{(\pm 1, 0), (0, \pm 1), (1, \pm 1), (-1, \pm 1)\}$ y se muestra en la figura 5.1.

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	(i, j)	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

Figura 5.1. Representación espacial de sistema de vecindad N^2 .

La energía de una imagen I en el valor de intensidad l ($0 \leq l \leq L$) se calcula generando una matriz bidimensional para cada valor de intensidad. Así $B_l = \{b_{i,j}, 1 \leq i \leq m, 1 \leq j \leq n\}$ donde $b_{i,j} = 1$ si el valor de intensidad en la posición actual es mayor que l ($I_{i,j} > l$), en caso contrario $b_{i,j} = -1$. La figura 2 muestra la imagen de escala de grises I con los valores de intensidad l en cada píxel. También se muestran dos ejemplos de la matriz binaria B_l . Tomando la figura 5.2 como ejemplo, sea $C = \{c_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ una matriz constante tal que $c_{ij} = 1, \forall (i, j)$, entonces el valor de la energía E_l de la imagen I en el valor de intensidad l se determina por:

$$E_l = - \sum_{i=1}^m \sum_{j=1}^n \sum_{pq \in N_{ij}^2} b_{ij} \cdot b_{pq} + \sum_{i=1}^m \sum_{j=1}^n \sum_{pq \in N_{ij}^2} c_{ij} \cdot c_{pq} \quad (5.1)$$

El segundo sumando de la Ec. 5.1 es un término constante cuya finalidad es asegurarse que el resultado sea un valor positivo de la energía $E_l \geq 0$. Una inspección rápida a la Ec. 5.1 muestra que para una imagen dada I en el valor de intensidad l será cero si todos los elementos de la matriz binaria B_l son valores de 1 ó -1. Este método determina la energía asociada a cada nivel de intensidad de la imagen para generar una curva que considera la información del contexto espacial de la imagen.

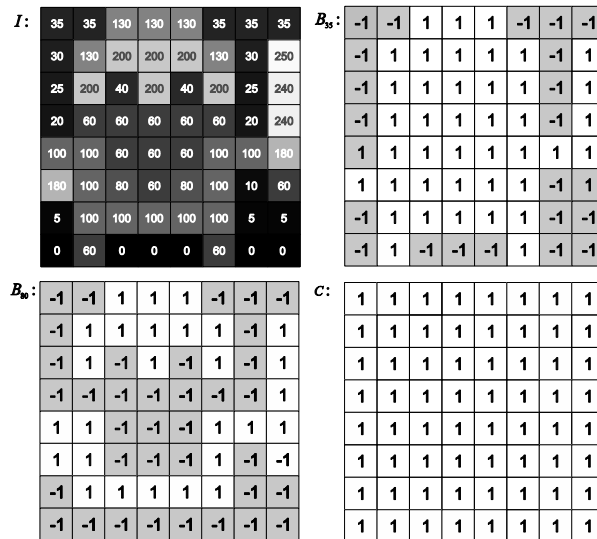


Figura 5.2. Imagen de ejemplo I , matriz binaria B_l para $l=35$ y $l=80$. Matriz constante

5.2 Segmentación de imágenes sensibles al contexto usando ALO o SCA.

Esta sección presenta los elementos básicos de los algoritmos *antlion optimizer* (ALO) y *sine cosine algorithm* (SCA) y cómo fueron modificados para realizar la umbralización multinivel. Los algoritmos evolutivos seleccionados exploran el espacio de búsqueda definido por la curva de energía y no por el histograma. Para evaluar los efectos de la curva de energía en el proceso de umbralización se utilizarán dos de los criterios paramétricos ampliamente usados en la comunidad científica sobre segmentación de imágenes, a saber: el propuesto por Otsu (1979) con base la varianza y el propuesto por Kapur et al. (1985) con base en la entropía. Así, cada algoritmo descrito en este capítulo se evalúa frente a los métodos de Otsu y Kapur.

En la Figura 5.3 se muestra la metodología de forma gráfica. La imagen I se incorpora al proceso para después determinar su curva de energía. Después se aplican los pasos principales de los algoritmos ALO o SCA para encontrar los mejores valores de umbral que se requieren para segmentar dicha curva de energía. Así, el algoritmo se repetirá hasta que se alcance un criterio de parada.

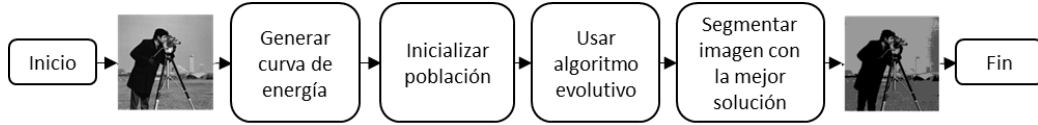


Figura 5.3. Metodología propuesta para la incorporación de información contextual.

El primer paso para cada algoritmo (ALO/SCA) es la generación de una población aleatoria que representa los valores de umbral para una imagen dada. Cada elemento de la población contendrá el conjunto de umbrales candidatos.

Además, de forma similar a la mayoría de algoritmos evolutivos, las soluciones candidatas se generan dentro de un espacio de búsqueda definido por los límites $[g_{\min}, g_{\max}]$, donde g_{\min} y g_{\max} son los valores mínimos y máximos que puede tomar la curva de energía respectivamente. Así, cada variable de decisión th_i de la i -ésima solución (**TH**) puede generarse usando la Ec. 5.2:

$$th_i = g_{\min} + rand \times (g_{\max} - g_{\min}), \quad \forall i = [1, 2, \dots, K] \quad (5.2)$$

donde $rand$ es un número aleatorio uniformemente distribuido entre 0 y 1; th_i es el i -ésimo valor de umbral de la solución **TH** _{i} donde i corresponde a una dimensión del espacio de búsqueda.

A continuación, el algoritmo ALO o SCA se utiliza para actualizar las soluciones más recientes de acuerdo con los comportamientos descritos en cada algoritmo hasta que se alcanza

alguna condición de parada. Cuando el algoritmo llega a su fin el proceso finaliza, proporcionando el mejor conjunto de umbrales encontrados para después segmentar la imagen usando la Ec. 2.6. Las secciones siguientes explican los componentes principales de los algoritmos ALO y SCA.

5.2.1 Optimizador de la hormiga león (ALO) para UM

El objetivo de esta subsección es describir al algoritmo de optimización de la hormiga león (ALO) aplicado a la determinación de la segmentación multinivel. El algoritmo ALO está inspirado en el comportamiento de las hormigas león. Tales insectos tienen comportamientos de caza peculiares, ya que crean trampas en forma de pozos donde esperan hasta que su presa (hormigas normales) cae. Para facilitar que la presa caiga en la trampa, las hormigas león lanzan arena a las presas para que resbalen y puedan ser atrapadas.

Desde el punto de vista computacional se crea una población original \mathbf{X} de N individuos, para después evaluar la calidad de los umbrales propuestos por cada partícula mediante el uso del método de Otsu o Kapur. Después, en cada iteración el algoritmo ALO modifica la posición de cada individuo de acuerdo con un recorrido aleatorio descrito por la Ec. 5.3.

$$\mathbf{TH}_i^t = \frac{(\mathbf{TH}_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (5.3)$$

donde a_i y b_i son los valores mínimos y máximos de la caminata aleatoria para la i -ésima variable de decisión; c_i^t y d_i^t son los valores mínimos y máximos de la i -ésima variable de decisión en la iteración t . La Ec. 5.3 representa la caminata aleatoria normalizada y restringida a los límites del espacio de búsqueda.

El siguiente paso en el algoritmo ALO determina cómo la caminata aleatoria es afectada por las trampas de las hormigas león. Este comportamiento se describe mediante la siguiente expresión,

$$\begin{aligned} c_i^t &= Antlion_j^t + c^t \\ d_i^t &= Antlion_j^t + d^t \end{aligned} \quad (5.4)$$

donde c^t y d^t son los valores mínimos y máximos de todas las variables de decisión en el instante t . Mientras tanto, se usa un operador de ruleta para seleccionar la hormiga león j en la iteración t ($Antlion_j^t$). Este elemento es una solución candidata \mathbf{TH}_i extraída de la población \mathbf{X} . La Ec. 5.4 establece que la caminata aleatoria de las hormigas león está definida por una hipersfera definida por los vectores \mathbf{c} y \mathbf{d} alrededor de una hormiga león seleccionada donde

$\mathbf{c} = \{c_1^t, c_2^t, \dots, c_i^t\}$ y $\mathbf{d} = \{d_1^t, d_2^t, \dots, d_i^t\}$. Basados en este concepto, las hormigas león están restringidas a moverse dentro del espacio delimitado por la hiper-esfera que tiene por centro una hormiga. El algoritmo ALO construye la estrategia para cazar hormigas usando el operador de ruleta, el cual se utiliza para la selección de una hormiga león (dado que las hormigas caen sólo en una trampa de las hormigas león) basados en los valores de la función de aptitud. Siguiendo esta estrategia, las hormigas león con mejor valor de aptitud tienen mayor probabilidad de atrapar hormigas comunes.

Al concluir la etapa de construcción de trampas, las hormigas comunes se deslizan hasta caer junto a una hormiga león, la cual lanza arena desde el centro del pozo para hacer que la hormiga caiga más rápido. Para imitar este comportamiento, el radio de cada hiper-esfera se disminuye adaptativamente usando la siguiente ecuación:

$$c^t = \frac{c^t}{rad}, \quad d^t = \frac{d^t}{rad}, \quad rad = \frac{10^w t}{T} \quad (5.5)$$

donde rad es el radio de explotación, w es una constante definida de acuerdo con la iteración ($w=2$ cuando $t < 0.1T$, $w=3$ cuando $t < 0.5T$, $w=4$ cuando $t < 0.75T$, $w=5$ cuando $t < 0.9T$ y $w=10$ cuando $t < 0.95T$). En otras palabras, w ajusta la intensidad de explotación. Después, en la etapa final el objetivo es atrapar la presa cuando alcanza el fondo del pozo. En este paso, la hormiga león se retira para que la hormiga pueda ser capturada. Para formular este proceso matemáticamente se utilizan la siguiente ecuación

$$Antlion_j^t = Ant_i^t \quad \text{si} \quad f(Ant_i^t) > f(Antlion_j^t) \quad (5.6)$$

En la Ec. 5.6 $Antlion_j^t$ y Ant_j^t son soluciones \mathbf{TH}_i extraídas de \mathbf{X} en la iteración t . Esta ecuación considera que cuando una hormiga Ant_j^t tiene mejor valor de aptitud que la hormiga león $Antlion_j^t$, la hormiga es cazada por la hormiga león cambiando esta última su posición a la de la presa.

El concepto de elitismo se aplica en este algoritmo para mantener la mejor solución obtenida en cada iteración y a su vez afectar los movimientos del resto de hormigas. En este sentido, cada hormiga camina aleatoriamente alrededor de una hormiga león seleccionada (mediante el operador de la ruleta) y la hormiga de élite simultáneamente usando la Ec. 5.7.

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (5.7)$$

R'_A y R'_E son las caminatas aleatorias alrededor de la hormiga león seleccionada y la hormiga león de élite en cada iteración t respectivamente. Basado en todos los pasos previos de ALO, las dos caminatas aleatorias se usan para actualizar la posición de las hormigas. Además, los parámetros c y d se actualizan con respecto a la iteración actual de acuerdo con la Ec. 5.5. La función de aptitud se utiliza para determinar las hormigas presa y basado en esta evaluación las hormigas león actualizan su posición de acuerdo con lo siguiente: si alguna de las hormigas presa resulta ser más apta que cualquier otra hormiga león, entonces la posición de la hormiga león es desplazada a la posición de la hormiga presa en la siguiente ecuación. Entonces, la mejor hormiga león (de élite) se actualiza. Los pasos anteriores son repetidos hasta que se alcanza un criterio de paro previamente establecido.

Como se mencionó previamente, el algoritmo ALO intenta imitar los mecanismos de caza de la hormiga león. A diferencia de otros métodos basados en comportamientos de hormigas como el ACO o ACOR, en el algoritmo ALO la hormiga león es un depredador que genera una trampa para que lleguen a caer las hormigas presa. Además, en el algoritmo ACO la metáfora involucra la búsqueda por las mejores fuentes de comida mientras refuerzan la trayectoria generada mediante el uso de feromonas.

5.2.2 Algoritmo seno-coseno para UM

Para el problema de la umbralización multinivel, el algoritmo SCA tiene como objetivo encontrar soluciones óptimas en el dominio del espacio de búsqueda. En otras palabras, cada solución representa los valores de umbrales óptimos que maximizan los valores de las funciones objetivo usadas según Otsu (1979) y Kapur et al., (1985). La umbralización multinivel usando SCA utiliza como entrada la curva de energía de una imagen con una población inicial \mathbf{X} , de donde surgirá la mejor solución \mathbf{TH}_{best} después del proceso iterativo. En SCA, cada solución candidata \mathbf{TH}_i ($i=1,2,\dots, N$) es representada como un vector que contiene todos los posibles valores naturales correspondientes a los umbrales. La calidad de cada solución candidata se evalúa mediante la función objetivo correspondiente.

El algoritmo SCA tiene un mecanismo novedoso para actualizar la posición de las soluciones candidatas después de evaluar la función objetivo. Las Ec. 5.8 y Ec. 5.9 presentan el proceso para determinar las nuevas posiciones basados en las funciones matemáticas de seno y coseno (Mirjalili, 2016b).

$$th_j = th_j + r_1 \times \sin(r_2) \times \left| r_3 \times (th_{best}^j - th_j) \right|, \quad \forall j = [1, 2, \dots, nt] \quad (5.8)$$

$$th_j = th_j + r_1 \times \cos(r_2) \times \left| r_3 \times (th_j^{best} - th_j) \right|, \quad \forall j = [1, 2, \dots, nt] \quad (5.9)$$

En las Ec. 5.8 y Ec. 5.9, th_j es la solución actual en la variable de decisión j . Las variables aleatorias r_1, r_2 y r_3 se obtienen a partir de una distribución uniforme entre 0 y 1. Además $th_j^{best} \in \mathbf{TH}_{best}$ es la posición de destino y el operador $||$ indica el valor absoluto (Mirjalili, 2016b).

Basado en la formulación original del SCA (Mirjalili, 2016b), el proceso para determinar las siguientes áreas de la siguiente solución son responsabilidad de r_1 ; estas áreas pueden estar entre el espacio comprendido entre th_i y th_i^{best} , o pueden encontrarse fuera de este espacio. Este parámetro se actualiza durante las iteraciones de la siguiente forma:

$$r_1 = a - t \left(\frac{a}{T} \right), \quad (5.10)$$

donde t es la iteración actual. T es el número máximo de iteraciones y es constante. La determinación de la dirección de movimiento de la solución candidata \mathbf{TH}_i es responsabilidad del parámetro r_2 . El parámetro r_3 se utiliza para determinar el peso aleatorio para r_1 con el fin de des enfatizar estocásticamente $r_3 < 1$ o $r_3 > 1$.

Sin embargo, cada solución puede actualizarse simultáneamente de acuerdo con el valor de la variable aleatoria r_4 , la cual es usada para elegir entre el operador de seno o el operador de coseno. Por tanto, las ecuaciones Ec. 5.8 y Ec. 5.9 se combinan en la Ec. 5.11:

$$th_j = \begin{cases} th_j + r_1 \times \sin(r_2) \times |r_3 \times (th_j^{best} - th_j)|, & \text{si } r_4 < 0.5 \\ th_j + r_1 \times \cos(r_2) \times |r_3 \times (th_j^{best} - th_j)|, & \text{en otro caso} \end{cases}, \quad \forall j = [1, 2, \dots, nt] \quad (5.11)$$

En resumen, el funcionamiento del optimizador SCA requiere un primer paso donde el algoritmo SCA genera una población aleatoria y determina el valor de aptitud de cada individuo \mathbf{TH}_i . Después el algoritmo determina cuál es la mejor solución global y la aparta para ser considerada el punto objetivo \mathbf{TH}_{best} mientras que el resto de la población actualizándose actualiza de acuerdo con el punto objetivo \mathbf{TH}_{best} . Además, para enfatizar la explotación del espacio de búsqueda, los valores de los parámetros r_1, r_2, r_3 y r_4 se actualizan en cada iteración. El criterio de parada se establece como el máximo número de iteraciones T .

5.3 Resultados experimentales

Los algoritmos ALO y SCA aplicados a la umbralización multinivel de imágenes se evalúan sobre un conjunto de imágenes de prueba que contienen distintos niveles de complejidad. Los resultados completos de esta investigación se encuentran disponibles como

material complementario del artículo (Oliva, Hinojosa, Elaziz, & Ortega-Sánchez, 2018) donde se analizan los resultados numéricos sobre 11 imágenes. El estudio y análisis reflejado aquí se ha ampliado hasta completar un total de 56 imágenes, de las cuales se han seleccionado tres ejemplos significativos y representativos del conjunto total, que sirven para ilustrar tanto los resultados gráficos como numéricos. En la figura 5.4 se muestran estas tres imágenes junto a sus histogramas y curvas de energía. Se debe hacer notar que todas las imágenes mostradas presentan características y distribuciones distintas. En este contexto, la curva de energía preserva algunas características relativas a la distribución mencionada, incorporando además la información contextual de cada píxel. Un ejemplo claro de las diferencias existentes entre el histograma de una imagen y la curva de energía puede observarse claramente en la imagen Hunter, en la que el histograma presenta una distribución con tendencia plana tras un pico en la zona de valores bajos de intensidad, mientras que la curva de energía muestra un modo significativo en la zona central del rango de valores. En las otras dos imágenes las diferencias no son tan evidentes mostrando en ambos casos diferentes modos.

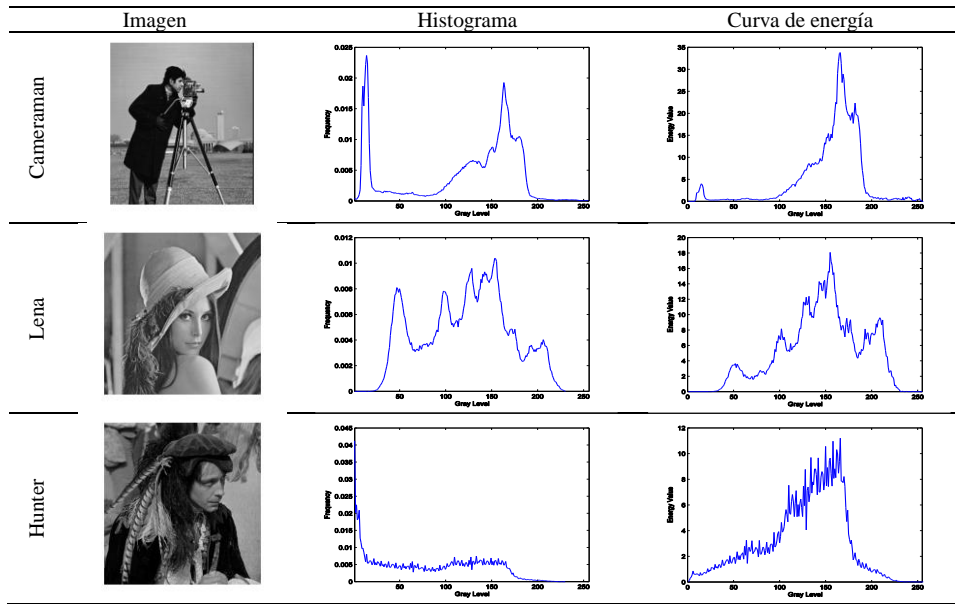


Figura 5.4. Subconjunto de imágenes de prueba y sus respectivos histogramas y curvas de energía.

Para verificar la eficacia de los enfoques propuestos se han comparado contra diferentes algoritmos metaheurísticos que forman parte de métodos clásicos y estado del arte. Entre los algoritmos usados se encuentra el *particle swarm optimization* (PSO) (Kennedy & Eberhart, 1995), *genetic algorithms* (GA) (Goldberg & Holland, 1988), *crow search algorithm* (CSA) (Askarzadeh, 2016), *runner root algorithm* (RRA) (Merrikh-Bayat, 2015) y *ant colony optimization for continuous domains* (ACOR) (Socha & Dorigo, 2008). Todos los algoritmos metaheurísticos seleccionados están basados en las recomendaciones propuesta por los autores de los mismos. De acuerdo al teorema NFL, no todos los algoritmos de optimización pueden

aplicarse eficientemente al mismo problema (Wolpert & Macready, 1997). En este contexto, los experimentos y comparaciones entre los algoritmos metaheurísticos aplicados a la umbralización pueden ayudar a determinar si los algoritmos ALO y SCA son competitivos para aplicaciones del mundo real.

Considerando que los algoritmos metaheurísticos contienen variables aleatorias, es necesario analizar estadísticamente sus resultados. Para ello se han llevado a cabo 35 pruebas independientes para cada experimento realizado. La segmentación se realiza considerando 2, 4, 8, 16 y 32 umbrales nt sobre la curva de energía. El criterio de paro y el tamaño de población para cada algoritmo es de 500 y 50 respectivamente para cada algoritmo. Todos los experimentos fueron realizados usando MATLAB 8.3 (MathWorks, 2018) en un procesador Intel Xeon-2620 v3 CPU @ 2.4Ghz con 16GB de memoria RAM.

5.3.1 Resultados usando la varianza entre clases

En esta subsección se presentan los resultados obtenidos por los algoritmos ALO y SCA usando la formulación de Otsu (1979). Para ello ALO y SCA usan la Ec. 2.14 como función objetivo con el fin de maximizar la varianza entre clases. Los resultados se obtienen sobre 21 imágenes con cinco números de umbrales distintos. La Tabla 5.1 presenta una tabla con los mejores umbrales obtenidos de acuerdo a su valor de aptitud por los distintos métodos de la umbralización. En la Tabla 5.1, la columna I corresponde al nombre de la imagen, la columna nt indica el número de umbrales usados ($nt=2, 4, 8, 16, 32$). El resto de las columnas corresponden a los umbrales obtenidos por los algoritmos ALO, SCA, GA, PSO, CSA, RRA y ACOR. Tales umbrales pueden ser aplicados directamente sobre la curva de energía.

Tabla 5.1. Umbrales obtenidos por los algoritmos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando el método de Otsu como función objetivo.

I	nt	ALO	SCA	GA	PSO	CSA	RRA	ACOR
Cameraman	2	83 153	83 153	84 153	87 151	83 153	83 153	83 153
	4	78 141 168 202	79 140 168 201	71 140 169 205	70 143 159 177	78 141 168 202	78 141 168 202	72 141 165 195
	8	40 86 120 140	40 77 124 140	36 75 106 125	47 81 126 158	38 84 118 139	38 84 118 139	37 78 109 128
		156 168 179 207	46 163 174 199	146 162 176	177 204 234 236	156 168 179 208	156 168 179 208	144 160 175 205
	16	26 51 78 100 114	27 27 42 79 80	1 25 40 48 69	21 41 42 45 66	24 53 80 104	24 53 80 104 119	23 39 56 76 97
		125 135 145 154	103 113 133 144	90 113 133 147	90 99 114 132	119 131 141 150	131 141 150 158	111 123 133 143
		161 168 175 182	158 167 173 176	156 168 174	139 155 164 175	158 165 171 178	165 171 178 184	152 160 168 176
		193 211 230	183 189 215	182 203 229	192 198 204	184 193 208 228	193 208 228	183 197 222
	32	15 23 38 52 65	4 30 31 49 51 67	1 11 11 13 21	31 39 43 46 46	14 23 29 35 41	14 23 29 35 41	1 1 12 16 23 32
		75 85 97 105 113	70 77 91 110	34 41 61 87 92	56 57 61 76 85	56 72 82 92 105	56 72 82 92 105	42 52 62 73 85 96
		121 127 133 139	111 114 117 118	94 107 113 118	93 96 103 105	113 120 127 133	113 120 127 133	105 112 119 126
		145 150 154 157	129 134 141 144	126 130 139	113 118 132 138	140 146 153 158	140 146 153 158	133 140 146 153
		161 164 168 171	148 151 160 167	144 148 155	140 147 158 162	162 166 170 174	162 166 170 174	158 163 167 172
		175 179 183 187	173 177 179 185	157 165 170	166 167 174 175	177 179 183 185	177 179 183 185	177 181 185 191
		196 206 216 229	185 187 190 212	178 185 190	180 188 209 211	188 193 205 215	188 193 205 215	199 209 221 235
		236 242	232 241	201 201 212	234 255	222 235	222 235	
				226 233 242				
Lena	2	114 169	114 169	114 170	115 168	114 169	114 169	114 169
	4	80 119 149 183	78 118 147 182	83 121 150 189	72 100 135 186	80 119 149 183	80 119 149 183	80 119 149 183
	8	67 92 114 134	65 73 96 115	71 95 117 137	3 51 97 120 150	63 90 113 133	63 90 113 133	63 86 105 121
		150 166 184 203	140 160 179 192	154 176 202	163 178 203	149 165 184 203	149 165 184 203	137 153 172 195
				237				
	16	48 58 72 85 97	51 80 91 93 105	20 53 70 84 92	3 23 54 57 72 91	50 61 74 90 104	50 61 74 90 104	46 55 65 78 90
		109 121 132 142	118 133 135 151	108 129 145	106 120 133 149	116 126 135 145	116 126 135 145	101 111 122 132
		151 160 170 182	159 166 179 186	156 169 180	169 174 183 197	154 167 181 195	154 167 181 195	142 151 160 171
		193 203 213	188 195 212	189 196 207	239 246	205 215 243	205 215 243	184 197 209
				222 248				
	32	7 36 46 51 56 66	13 35 53 53 61	1 1 11 19 24 33	5 20 25 26 29 39	3 28 31 49 56 64	3 28 31 49 56 64	1 1 37 43 47 51

		74 83 90 95 101	68 73 84 93 103	45 53 57 64 83	44 48 52 53 53	70 75 81 87 96	70 75 81 87 96	55 61 67 75 83 90
		105 111 116 122	103 111 119 125	89 102 103 113	58 63 69 79 97	104 111 118 124	104 111 118 124	97 104 111 119
		127 133 139 145	138 142 151 164	121 130 138	103 114 115 123	129 133 140 146	129 133 140 146	126 132 139 145
		150 155 160 167	166 167 171 171	145 153 162	133 136 149 151	151 156 160 165	151 156 160 165	151 156 161 167
		174 181 189 195	174 185 188 197	170 179 183	162 177 181 190	170 175 181 187	170 175 181 187	173 178 184 191
		200 205 210 217	202 207 220 245	193 199 207	201 209 215 251	195 202 208 215	195 202 208 215	197 203 209 217
		241	250 255	213 217 229		226	226	
				251 252				
Hunter	2	87 141	87 141	87 140	87 142	87 141	87 141	87 141
	4	56 96 130 160	56 97 132 161	49 93 128 158	53 106 131 154	56 96 130 160	56 96 130 160	56 96 130 160
	8	38 65 89 110 128	30 67 93 115	46 68 95 120	54 79 114 128	42 71 97 118	42 71 97 118 135	34 60 83 105 124
		145 161 184	135 146 163 204	138 156 183	152 187 214 247	135 151 166 187	151 166 187	142 160 183
	16	20 36 51 65 79	1 8 26 39 53 64	16 36 59 73 89	10 18 18 42 53	24 44 60 75 88	24 44 60 75 88	18 33 46 59 72 85
		93 105 116 125	77 90 102 112	94 102 112 131	70 73 93 115 122	100 109 117 124	100 109 117 124	99 110 121 131
		134 143 152 160	124 133 142 150	140 153 155	129 141 156 176	131 141 151 160	131 141 151 160	141 151 160 169
		169 181 199	171 193	171 181 187	187 230	170 183 201	170 183 201	182 200
	32	1 5 12 20 30 41	14 24 40 49 55	2 17 30 34 36	28 41 56 66 83	8 14 20 31 40 49	8 14 20 31 40 49	1 9 17 24 33 41
		48 56 64 71 78	58 59 61 61 74	41 55 66 80 95	102 116 119 125	57 64 71 77 83	57 64 71 77 83	49 58 66 74 82 90
		86 93 100 105	87 94 94 98 108	102 103 112	128 130 132 139	89 97 105 110	89 97 105 110	98 106 113 119
		111 117 123 129	118 126 127 130	119 125 130	144 153 154 160	115 120 126 131	115 120 126 131	125 131 137 142
		135 141 146 152	131 137 138 147	137 148 154	166 181 192 196	139 148 156 165	139 148 156 165	147 152 157 162
		158 162 167 173	153 157 160 165	159 160 167	197 202 206 212	170 176 184 192	170 176 184 192	167 172 177 184
		180 189 199 208	166 171 184 201	175 183 197	213 228 234 235	200 207 214 223	200 207 214 223	191 199 209 221
		221	252	202 210 212	236 237 240	234	234	
				219 231 237				
				244				

El objetivo de la segmentación multinivel es obtener la mejor calidad posible en las imágenes segmentadas mediante el uso de umbrales. En este contexto, los resultados del PSNR entre la imagen segmentada y la imagen original deben analizarse. Para comparar estadísticamente las soluciones obtenidas por los distintos enfoques, se calcula la media del valor de aptitud generado por la función objetivo y su desviación estándar (σ). Los valores correspondientes a este análisis se muestran en la Tabla 5.2.

Tabla 5.2. Media y σ de los valores de PSNR obtenidos por los algoritmos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando Otsu como función objetivo.

I	nt	ALO		SCA		GA		PSO		CSA		RRA		ACOR	
		Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ
Cameraman	2	17.6057	7.20E-15	17.6038	5.70E-02	17.6129	2.40E-02	17.6048	1.20E-01	13.8636	7.20E-15	17.6057	7.20E-15	17.3993	1.55E-01
	4	19.4286	7.20E-15	19.3005	4.70E-01	19.3774	1.80E-01	19.7264	8.10E-01	19.1721	1.50E-02	19.4246	1.80E-02	19.4852	7.11E-01
	8	25.4905	2.50E-02	24.3788	8.50E-01	25.587	4.50E-01	25.0233	9.10E-01	25.0992	2.10E-01	25.5142	3.10E-01	25.5398	1.49E-01
	16	29.7947	3.60E-01	27.9076	8.20E-01	29.356	6.10E-01	28.7481	7.90E-01	29.5755	2.40E-01	29.4064	5.40E-01	29.6193	5.92E-02
	32	35.5253	1.60E+00	32.3933	1.10E+00	33.2438	1.50E+00	32.0634	1.50E+00	37.4424	1.50E+00	34.3886	2.00E+00	35.6078	4.43E-01
Lena	2	13.3913	3.60E-15	13.41	3.70E-02	13.3985	2.10E-02	13.3956	4.00E-02	10.5764	3.60E-15	13.3913	3.60E-15	13.1654	5.26E-02
	4	18.3293	7.20E-15	18.1998	4.90E-01	18.3545	1.60E-01	18.3399	2.30E-01	16.8339	1.80E-03	18.3294	4.20E-04	17.9957	5.73E-02
	8	20.9123	1.50E-01	21.0618	1.60E+00	21.4104	1.10E+00	20.7929	1.10E+00	20.7413	9.60E-02	21.1996	9.90E-01	20.8341	1.29E-01
	16	26.8999	2.20E+00	25.5527	2.20E+00	27.464	1.90E+00	25.0613	2.80E+00	27.0567	1.10E+00	25.5768	2.60E+00	26.2025	5.43E-01
	32	34.9573	1.90E+00	31.7432	1.20E+00	33.1823	1.70E+00	30.633	3.60E+00	37.3675	1.90E+00	33.8087	2.70E+00	33.3224	5.87E-01
Hunter	2	17.1165	1.10E-14	17.1254	2.20E-02	17.1183	5.70E-03	17.1105	3.20E-02	14.2921	1.10E-14	17.1166	1.10E-14	16.8552	8.47E-02
	4	21.3132	7.20E-15	21.079	7.70E-01	21.3513	1.70E-01	21.0799	5.90E-01	18.752	2.00E-02	21.3332	4.60E-02	20.9367	5.61E-02
	8	25.6913	1.50E-01	24.3513	8.40E-01	25.4032	7.60E-01	24.4137	9.70E-01	24.3898	2.40E-01	25.6086	5.10E-01	25.5616	1.16E-01
	16	31.1252	3.10E-01	28.6971	9.70E-01	29.9834	6.50E-01	28.902	9.80E-01	30.9959	6.80E-01	30.7786	7.80E-01	31.068	1.48E-01
	32	36.6525	4.80E-01	32.809	8.50E-01	34.519	7.00E-01	32.9076	1.20E+00	37.0924	7.20E-01	35.8073	8.70E-01	36.5639	1.56E-01

En la Tabla 5.2, un valor mayor en la media del PSNR representa una mejor segmentación. Además, se espera que un valor menor de σ refleje una menor variabilidad entre los resultados obtenidos por cada experimento. Un estudio similar es presentado en la Tabla 5.3 donde se analiza el comportamiento de la métrica SSIM descrita en la sección 2.5.3, la cual tiene como objetivo analizar cómo las estructuras de la imagen son afectadas por la segmentación, un valor mayor indica una mejor segmentación.

Tabla 5.3. Media y σ de los valores de SSIM obtenido por los algoritmos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando Otsu como función objetivo.

Table 7: PSOR using different algorithms for different test images.															
nt	ALO		SCA		GA		PSO		CSA		RRA		ACOR		
	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	
Cameraman	2	0.7544	3.40E-16	0.7541	1.10E-03	0.7545	3.70E-04	0.7542	2.30E-03	0.5263	3.40E-16	0.7544	3.40E-16	0.7401	2.67E-03
	4	0.7789	1.10E-16	0.7776	6.80E-03	0.7782	2.70E-03	0.7858	1.60E-02	0.6212	2.10E-04	0.7789	3.40E-04	0.7706	1.21E-02
	8	0.8808	4.20E-04	0.879	1.90E-02	0.8831	5.50E-03	0.8767	1.80E-02	0.7439	2.80E-03	0.8808	3.70E-03	0.87	2.46E-03
	16	0.9236	9.20E-03	0.9289	1.90E-02	0.928	1.80E-02	0.9177	1.70E-02	0.8218	5.60E-03	0.9196	8.30E-03	0.906	6.42E-04

	32	0.9768	1.70E-02	0.9628	1.90E-02	0.9603	2.10E-02	0.9465	2.30E-02	0.9719	1.50E-02	0.9644	2.30E-02	0.9625	3.22E-03
Lena	2	0.5439	0.00E+00	0.5451	2.80E-03	0.5444	1.60E-03	0.5442	2.80E-03	0.2945	0.00E+00	0.5439	0.00E+00	0.5325	2.06E-03
	4	0.7475	4.50E-16	0.7424	1.70E-02	0.7484	6.50E-03	0.7475	9.90E-03	0.596	3.10E-04	0.7475	9.90E-05	0.733	1.09E-03
	8	0.8311	4.60E-03	0.829	4.10E-02	0.8389	2.50E-02	0.8225	2.80E-02	0.7522	2.90E-03	0.8362	2.20E-02	0.823	3.55E-03
	16	0.9351	2.40E-02	0.9148	3.40E-02	0.939	2.30E-02	0.9058	3.90E-02	0.8951	1.30E-02	0.9174	3.10E-02	0.914	5.72E-03
	32	0.9858	7.70E-03	0.973	5.90E-03	0.9788	9.70E-03	0.9613	2.60E-02	0.9799	1.10E-02	0.9804	1.40E-02	0.9651	2.20E-03
Hunter	2	0.6558	2.30E-16	0.6561	1.40E-03	0.6558	2.10E-04	0.6554	1.80E-03	0.3002	2.30E-16	0.6558	2.30E-16	0.644	1.30E-03
	4	0.8082	3.40E-16	0.8036	2.90E-02	0.8111	8.60E-03	0.8028	2.50E-02	0.4687	8.20E-04	0.8092	2.20E-03	0.7928	1.01E-03
	8	0.9099	3.60E-03	0.9053	3.00E-02	0.9091	2.40E-02	0.883	2.90E-02	0.6712	6.20E-03	0.9094	1.30E-02	0.9015	3.30E-03
	16	0.9713	2.80E-03	0.9652	1.20E-02	0.9681	9.10E-03	0.9528	1.60E-02	0.8469	7.40E-03	0.9688	8.60E-03	0.9564	1.62E-03
	32	0.9929	2.00E-03	0.9833	6.40E-03	0.9882	3.60E-03	0.9779	8.30E-03	0.9571	3.00E-03	0.9889	4.20E-03	0.9745	6.53E-04

Otra métrica interesante que ha sido utilizada para verificar la calidad de una imagen segmentada es FSIM descrita en la sección 2.5.4. La Tabla 5.4 muestra los resultados de FSIM usando los resultados obtenido por cada uno de los algoritmos metaheurísticos seleccionados. La métrica de FSIM indica una mayor calidad cuando alcanza un valor mayor.






Tabla 5.4. Media y σ de los valores de FSIM obtenidos por los algoritmos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando Otsu como función objetivo.

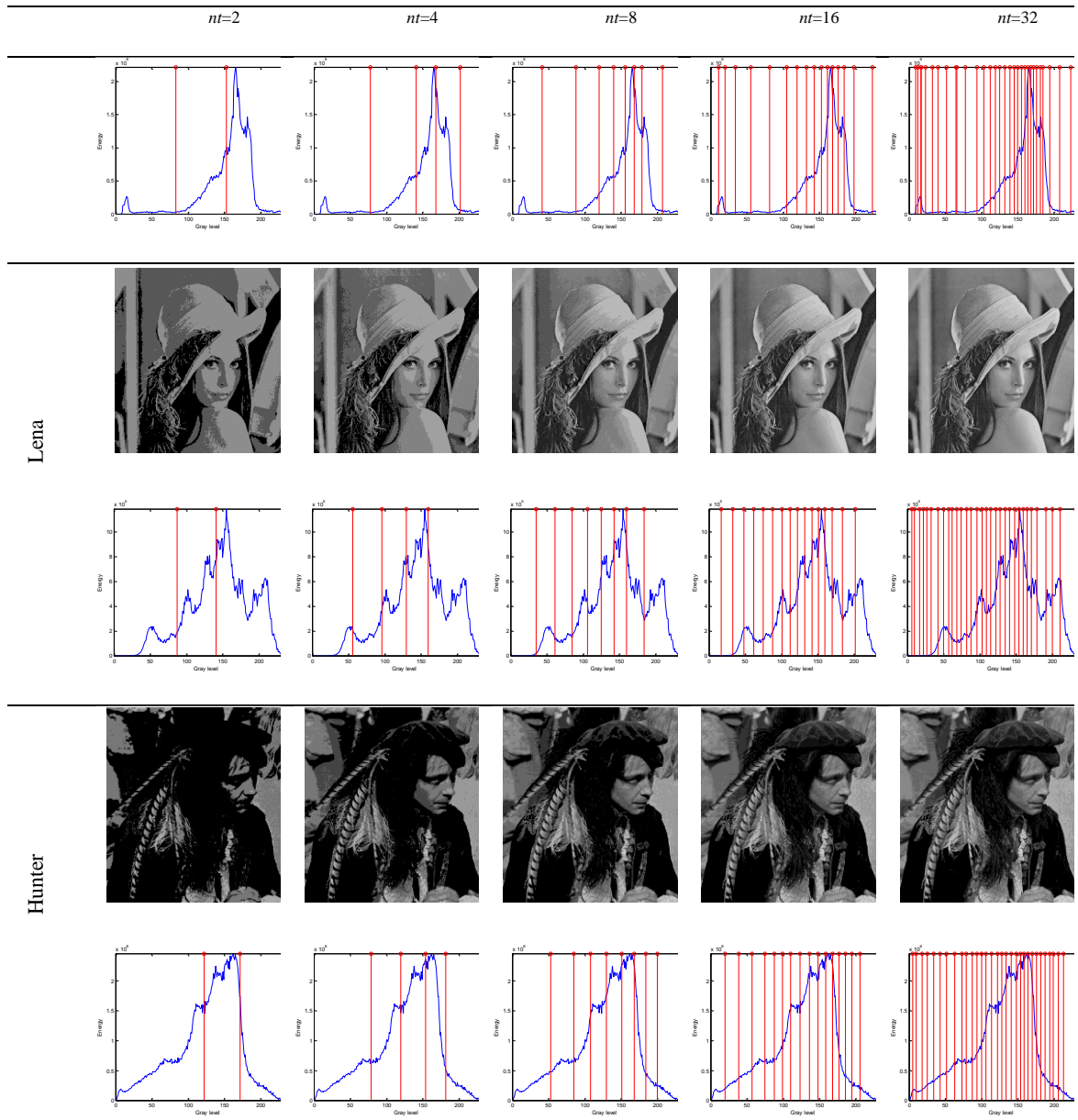
I	nt	ALO		SCA		GA		PSO		CSA		RRA		ACOR	
		Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ
Cameraman	2	0.7479	3.40E-16	0.7477	1.10E-03	0.748	3.70E-04	0.7478	2.30E-03	0.6749	5.60E-16	0.7479	5.60E-16	0.7358	2.80E-03
	4	0.7822	1.10E-16	0.7835	6.80E-03	0.7826	2.70E-03	0.7926	1.60E-02	0.7824	1.50E-04	0.7822	1.10E-04	0.7807	2.02E-02
	8	0.9158	4.20E-04	0.8948	1.90E-02	0.918	5.50E-03	0.906	1.80E-02	0.9105	2.90E-03	0.9164	3.90E-03	0.9056	2.88E-03
	16	0.9682	9.20E-03	0.9385	1.90E-02	0.9575	1.80E-02	0.953	1.70E-02	0.9668	2.40E-03	0.9638	5.80E-03	0.9524	4.28E-04
	32	0.9838	1.70E-02	0.9702	1.90E-02	0.9775	2.10E-02	0.9732	2.30E-02	0.9873	1.90E-03	0.9825	3.10E-03	0.9637	5.81E-04
Lena	2	0.6609	0.00E+00	0.6608	2.80E-03	0.6611	1.60E-03	0.6609	2.80E-03	0.5961	2.30E-16	0.6609	2.30E-16	0.6499	3.44E-03
	4	0.7974	4.50E-16	0.7895	1.70E-02	0.7973	6.50E-03	0.7959	9.90E-03	0.7391	8.50E-05	0.7974	3.30E-05	0.7825	2.28E-03
	8	0.8757	4.60E-03	0.851	4.10E-02	0.8719	2.50E-02	0.8617	2.80E-02	0.8668	2.20E-03	0.8761	7.40E-03	0.8607	1.47E-03
	16	0.933	2.40E-02	0.9057	3.40E-02	0.9274	2.30E-02	0.9112	3.90E-02	0.9333	2.80E-03	0.9225	1.50E-02	0.9024	3.72E-03
	32	0.9802	7.70E-03	0.9599	5.90E-03	0.9706	9.70E-03	0.9539	2.60E-02	0.9903	1.20E-02	0.9741	1.60E-02	0.9596	3.07E-03
Hunter	2	0.7615	2.30E-16	0.7616	1.40E-03	0.7615	2.10E-04	0.7611	1.80E-03	0.6477	3.40E-16	0.7615	3.40E-16	0.7512	5.06E-03
	4	0.8759	3.40E-16	0.8705	2.90E-02	0.877	8.60E-03	0.8711	2.50E-02	0.8119	4.10E-04	0.8764	1.10E-03	0.8601	2.38E-03
	8	0.9486	3.60E-03	0.9351	3.00E-02	0.9466	2.40E-02	0.9287	2.90E-02	0.9299	3.40E-03	0.9485	7.30E-03	0.9353	2.03E-03
	16	0.9856	2.80E-03	0.9752	1.20E-02	0.9822	9.10E-03	0.9722	1.60E-02	0.9852	4.30E-03	0.9838	5.10E-03	0.9684	9.15E-04
	32	0.9968	2.00E-03	0.9889	6.40E-03	0.9928	3.60E-03	0.9872	8.30E-03	0.9968	1.80E-03	0.9947	2.30E-03	0.9775	2.68E-04

En resumen, las métricas presentadas en las Tablas 5.2, 5.3 y 5.4 se utilizan para analizar la calidad de las imágenes segmentadas usando los mejores umbrales encontrados por cada uno de los métodos comparados. Los valores de PSNR, SSIM y FSIM observados indican que el algoritmo ALO puede encontrar mejores umbrales en comparación con las otras metodologías analizadas.

Las Tablas 5.5 y 5.6 muestran las imágenes obtenidas por los algoritmos ALO y SCA con distintos números de umbrales usando Otsu como función objetivo. En concreto, la Tabla 5.5 está basada en los mejores umbrales encontrados por el algoritmo ALO. Las imágenes resultantes hacen evidente cómo el incremento de número de umbrales aumenta significativamente la calidad de las imágenes, sobre todo desde el punto de vista cualitativo. Obsérvese en este sentido las imágenes con $nt = 32$.

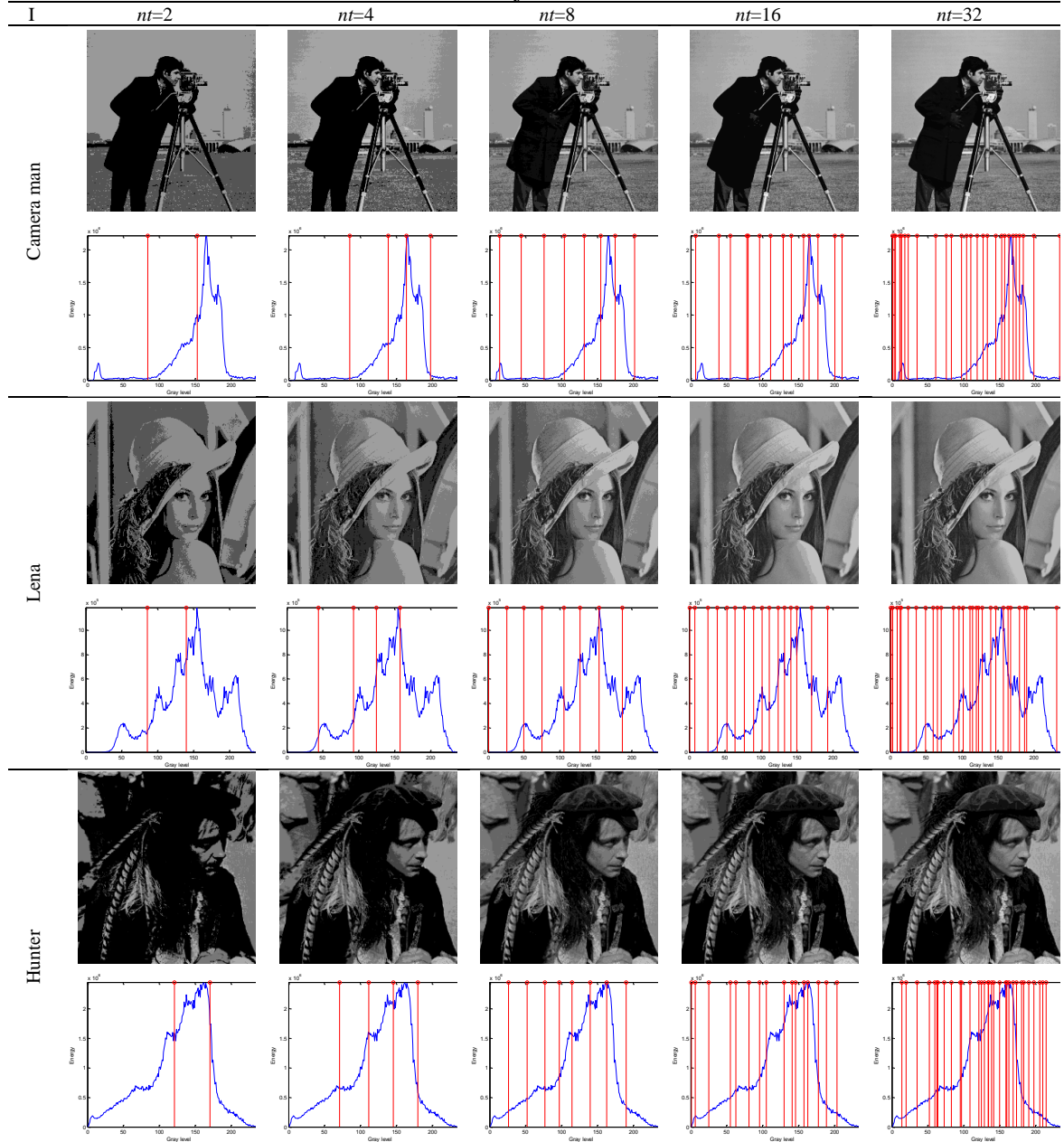
Tabla 5.5. Imágenes segmentadas y su curva de energía umbralada usando el algoritmo ALO con la función objetivo propuesta por Otsu.

	$nt=2$	$nt=4$	$nt=8$	$nt=16$	$nt=32$
Cameraman					



De forma similar a los resultados obtenidos por ALO, las imágenes de la Tabla 5.6 han sido obtenidas a partir de la umbralización multinivel mediante el algoritmo SCA en combinación con Otsu. Esta tabla también muestra las respectivas curvas de energía y los umbrales (marcados sobre ellas) encontrados por el algoritmo SCA. La calidad de las imágenes incrementa conforme al número de umbrales nt usados para cada segmentación. Obsérvese como en el caso anterior, la mejora de calidad con $nt = 32$. Sin embargo, cuantos más umbrales se utilizan, menor es la diferencia entre sus valores. En la Figura 5.5 es posible comparar el tiempo promedio que cada algoritmo utiliza donde cada línea vertical indica un segundo, donde el algoritmo ALO encuentra las soluciones en menor tiempo.

Tabla 5.6. Imágenes segmentadas y su curva de energía umbralada mediante el algoritmo SCA y la función objetivo de Otsu.



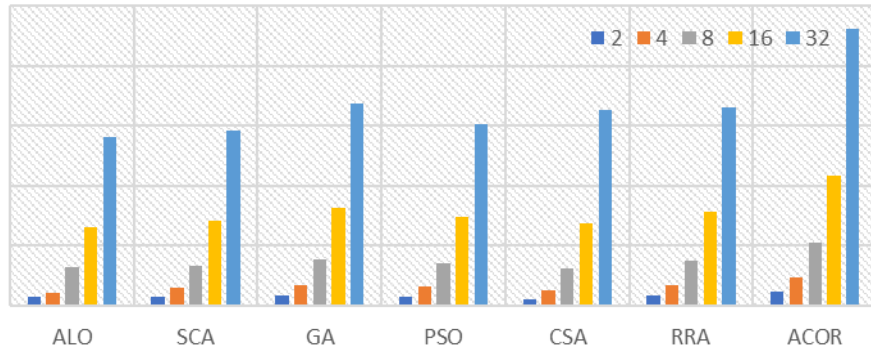


Figura 5.5. Tiempo promedio de cada algoritmo usando Otsu y 2, 4, 8, 16 y 32 umbrales.

5.3.2 Resultados usando la entropía máxima

En esta subsección se presentan los resultados de los algoritmos seleccionados para realizar umbralización multinivel usando como función objetivo la entropía propuesta por Kapur et al. (1985) (Ec. 2.22). El proceso de experimentación es el mismo que el de la subsección previa. En la Tabla 5.7 se presenta una comparación entre los umbrales obtenidos por cada uno de los algoritmos seleccionados (ALO, SCA, GA, PSO, CSA, RRA y ACOR).

Tabla 5.7. Umbrales encontrados por cada algoritmo (SCA, ALO, GA, PSO, CSA, RRA y ACOR) usando el método de Kapur como función objetivo.

I	nt	ALO	SCA	GA	PSO	CSA	RRA	ACOR
Camera man	2	107 196	107 196	107 196	109 198	107 196	107 196	107 196
	4	55 98 146 196	55 97 147 197	49 97 148 196	26 91 132 202	55 98 146 196	55 98 146 196	55 98 146 196
	8	22 47 72 98 126 157 193 222	26 57 85 106 130 163 193 213	21 42 65 94 121 155 192 216	26 46 60 81 107 173 202 213	22 50 78 102 129 158 193 222	22 46 69 94 121 150 193 222	23 53 83 107 134 160 193 222
	16	7 20 36 53 70 87 102 117 133 147 160 176 192 207 224 241	5 23 35 51 67 74 92 117 123 137 154 185 191 207 218 250	19 31 45 58 72 83 97 109 121 135 145 162 177 193 203 226	3 26 44 56 77 94 103 115 134 168 175 194 202 214 218 223	6 19 34 49 62 75 88 99 112 126 142 157 173 191 207 229	20 38 56 70 84 97 109 120 131 146 160 175 192 191 201 216 233	7 19 33 48 63 79 95 109 124 140 157 173 191 201 216 233
	32	7 17 23 34 43 50 59 67 75 84 92 102 109 115 122 129 135 141 147 154 160 166 176 183 191 200 207 215 222 230 236 244	6 11 17 23 27 39 41 49 60 77 84 88 91 99 113 121 131 150 159 165 185 187 189 198 202 208 217 221 225 230 240 250	1 7 17 21 28 39 45 53 59 66 74 87 95 99 107 111 115 123 133 145 151 155 168 176 187 197 207 215 224 231 237 243	13 18 24 28 29 31 40 44 51 52 52 55 66 74 89 96 107 112 115 127 128 141 165 167 182 192 195 207 213 216 225 234	7 13 19 25 31 37 45 52 58 65 71 78 85 94 102 110 121 132 138 147 162 169 179 189 197 201 205 211 217 225 233 242	7 19 26 33 39 44 51 60 68 76 84 91 98 107 115 123 129 136 143 149 156 163 171 181 191 201 210 218 224 229 235 243	7 12 18 24 30 37 43 50 57 64 71 79 87 95 103 112 118 124 132 141 149 157 165 173 182 191 198 205 214 224 233 243
Lena	2	94 163	94 163	94 163	89 161	94 163	94 163	94 163
	4	23 85 125 174	23 83 123 175	23 80 123 172	91 134 177 204	23 85 125 174	23 77 113 161	23 85 125 174
	8	23 58 85 112 140 169 197 245	19 51 76 88 117 151 188 245	23 43 62 85 109 133 165 194	19 82 112 132 140 160 187 212	23 62 89 114 137 159 180 202	23 58 87 113 137 159 179 201	23 60 86 111 133 154 177 201
	16	10 23 42 59 75 91 105 119 134 147 162 177 191 204 217 245	6 23 48 62 71 77 82 95 126 137 141 159 177 195 215 221	7 14 23 33 45 59 73 91 107 122 133 144 165 182 198 216	9 16 33 46 48 73 93 99 110 120 151 156 159 173 190 211	23 39 54 67 81 94 109 125 139 152 166 179 192 208 224 245	11 23 33 46 63 81 96 108 121 134 147 161 177 194 212 224	11 23 37 51 64 78 93 107 121 135 149 163 177 190 203 217
	32	4 10 16 23 33 40 47 54 61 69 77 88 95 102 107 114 119 125 131	5 8 22 24 29 34 44 47 52 54 67 67 72 79 84 98 100 107 119 127	1 4 10 15 20 31 40 44 52 57 67 73 83 90 102 108 117 124 131	2 6 17 17 28 42 45 49 56 71 81 95 99 106 117 118 129 146 152	13 20 32 38 44 49 55 63 70 77 83 93 103 112 122 129 136 143	5 11 17 23 27 32 37 43 49 55 60 66 72 78 84 90 97 104 111 127	2 6 12 17 23 33 39 45 53 61 69 77 86 94 102 109 117 126 135 143

		135 142 153	131 145 166	138 146 154	167 173 179	149 157 166	144 159 171	151 160 168
		158 165 173	170 175 182	158 164 171	189 201 204	177 188 201	179 188 194	175 183 190
		182 194 206	188 194 201	176 189 201	204 206 219	213 224 238	201 208 215	198 206 213
		218 223 230	212 216 247	207 212 219	223 236 244	241 244 245	223 230 245	219 230 245
		245		226	247	249 252		
Hunter	2	93 179	93 179	93 178	99 179	93 179	93 179	93 179
	4	45 93 133 179	48 92 132 180	40 89 132	32 82 114	45 93 133	41 89 131	45 93 133 179
				179	177	179	179	
	8	29 58 89 117	19 43 79 98	23 44 65 88	16 45 100	25 49 73 98	23 52 81 109	25 49 75 101
		145 174 198	128 157 188	114 145 183	153 181 205	125 152 182	140 175 197	127 154 183
		227	227	220	218 227	222	227	222
	16	17 32 46 61	13 27 40 53	9 16 29 41	16 26 30 52	13 26 41 57	18 35 53 69	13 24 37 51 65
		76 90 104 118	79 94 106 121	58 71 89	74 81 111	73 88 103	85 100 115	81 96 111 126
		132 148 163	140 152 161	109 125 137	134 140 162	118 133 148	130 144 157	141 156 171
		178 192 206	187 192 209	148 163 184	182 196 201	163 178 192	171 182 194	184 198 214
		220 233	225 240	195 215 229	203 218 222	206 221 233	207 220 234	230
	32	6 14 20 29 37	13 21 32 55	1 15 20 27	16 34 42 56	7 15 24 32 40	6 12 20 27 34	4 9 13 18 25
		46 53 61 70	57 65 74 84	38 44 49 54	66 71 75 76	48 56 64 72	41 50 60 66	31 37 44 51 59
		78 85 93 99	92 95 107 121	60 66 71 78	78 85 93 96	81 89 93 98	72 78 83 92	68 77 87 96
		105 114 121	128 135 141	90 97 106	101 110 117	105 113 121	101 108 115	104 111 120
		128 135 143	152 160 163	111 119 124	129 133 135	129 135 140	121 127 136	127 134 142
		151 159 167	169 183 187	130 138 151	141 147 148	144 150 156	146 157 165	149 157 166
		174 181 187	187 190 194	158 164 170	158 169 177	164 171 176	172 179 185	175 184 191
		195 202 208	205 209 215	175 186 194	193 203 205	183 189 197	191 198 204	199 208 214
		214 221 229	219 222 225	203 209 218	214 214 221	207 219 227	213 221 228	221 229 240
	39		233 242	223 235	222 236	234	236	

Los umbrales obtenidos por cada algoritmo metaheurístico se aplican a la curva de energía de cada imagen para clasificar los píxeles. Al realizar la clasificación se genera como resultado una imagen segmentada, de forma que con el fin de proceder a su análisis se evalúan distintas métricas de calidad. De forma similar a las secciones previas, la calidad de la segmentación se mide en términos de PSNR, SSIM y FSIM. En la Tabla 5.8 se presenta la media y también la desviación estándar (σ) de los valores de PSNR para todos los algoritmos, donde un valor mayor indica una mejor calidad en la segmentación.

Tabla 5.8. Media y σ de los valores de PSNR obtenidos por los métodos ALO, SCA, GA, PSO, CSA, RRA y ACOR usando el método de Kapur.

I	nt	ALO		SCA		GA		PSO		CSA		RRA		ACOR	
		Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ
Cameraman	2	14.1078	5.40E-15	14.1288	5.30E-02	14.1078	5.40E-15	14.1419	5.40E-02	13.9197	5.40E-15	14.0816	1.70E-01	13.8256	5.41E-15
	4	20.2737	1.90E-03	19.9807	3.40E-01	20.2755	3.30E-02	19.9816	7.10E-01	17.238	4.60E-03	20.2462	1.80E-01	19.8685	2.40E-03
	8	24.5888	5.50E-01	22.6049	1.20E+00	24.0827	4.20E-01	23.4598	1.60E+00	23.2151	4.40E-01	24.2921	5.90E-01	23.7074	3.59E-01
	16	29.3025	6.00E-01	26.676	1.80E+00	29.4739	7.00E-01	27.1448	1.80E+00	28.7052	8.40E-01	29.1105	8.40E-01	28.7225	7.12E-01
	32	34.674	1.20E+00	30.9462	1.80E+00	34.5137	8.90E-01	31.6333	1.60E+00	34.9571	9.10E-01	34.8934	1.10E+00	35.3253	2.98E-01
Lena	2	14.7275	4.70E-05	14.7324	4.20E-02	14.7275	9.00E-15	14.7292	5.20E-02	11.9276	9.00E-15	14.7275	3.30E-05	14.433	9.01E-15
	4	19.1959	2.60E-01	18.8979	2.80E-01	19.2272	4.80E-02	18.3044	7.90E-01	16.5724	3.60E-03	19.203	1.30E-01	18.7623	3.92E-01
	8	24.0628	5.80E-01	22.6521	8.80E-01	24.3503	2.90E-01	22.3626	1.50E+00	22.9017	2.80E-01	24.0068	4.90E-01	23.9697	1.91E-01
	16	29.9022	7.90E-01	26.3846	1.50E+00	29.6218	7.90E-01	26.5415	1.40E+00	29.1179	5.60E-01	29.6602	7.20E-01	29.5047	2.90E-01
	32	34.4498	1.30E+00	30.9993	1.30E+00	34.6621	6.10E-01	31.8827	1.40E+00	34.8237	5.40E-01	35.2422	8.00E-01	35.129	2.87E-01
Hunter	2	15.2267	2.10E-05	15.2209	2.20E-02	15.2259	4.50E-03	15.2172	2.20E-02	14.8311	1.20E-05	15.2268	3.50E-03	14.9222	1.99E-05
	4	21.126	1.90E-02	20.7969	1.70E-01	21.0681	4.20E-02	20.797	3.70E-01	18.2206	3.80E-02	21.0829	4.00E-02	20.6954	1.68E-02
	8	24.6765	1.90E-01	24.1929	1.00E+00	24.9561	6.10E-01	24.1361	1.30E+00	24.0139	5.50E-01	24.5576	2.40E-01	24.924	5.17E-01
	16	30.3006	2.30E-01	27.6459	1.40E+00	30.563	3.40E-01	27.5317	1.30E+00	29.6918	4.20E-01	30.2806	4.80E-01	30.2479	1.74E-01
	32	35.9282	4.70E-01	32.1482	1.20E+00	35.7301	4.30E-01	32.1929	1.10E+00	35.751	5.90E-01	35.8244	6.40E-01	36.0226	2.11E-01

En la Tabla 5.8 es posible observar que algunos algoritmos generan valores de PSNR altos. Sin embargo, esta métrica no puede asegurar por sí sola que la segmentación ha sido exitosa. Por ejemplo, algoritmos como GA o PSO obtienen umbrales malos, pero aun así en algunos casos logran obtener buenos valores de PSNR. Esto sucede ya que los píxeles pueden ser incorrectamente clasificados. Además, las tablas 5.9 y 5.10 muestran los resultados de las otras dos métricas que analizan la calidad de la segmentación.

Tabla 5.9. Media y σ de los valores de SSIM obtenidos por los métodos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando el método de Kapur.

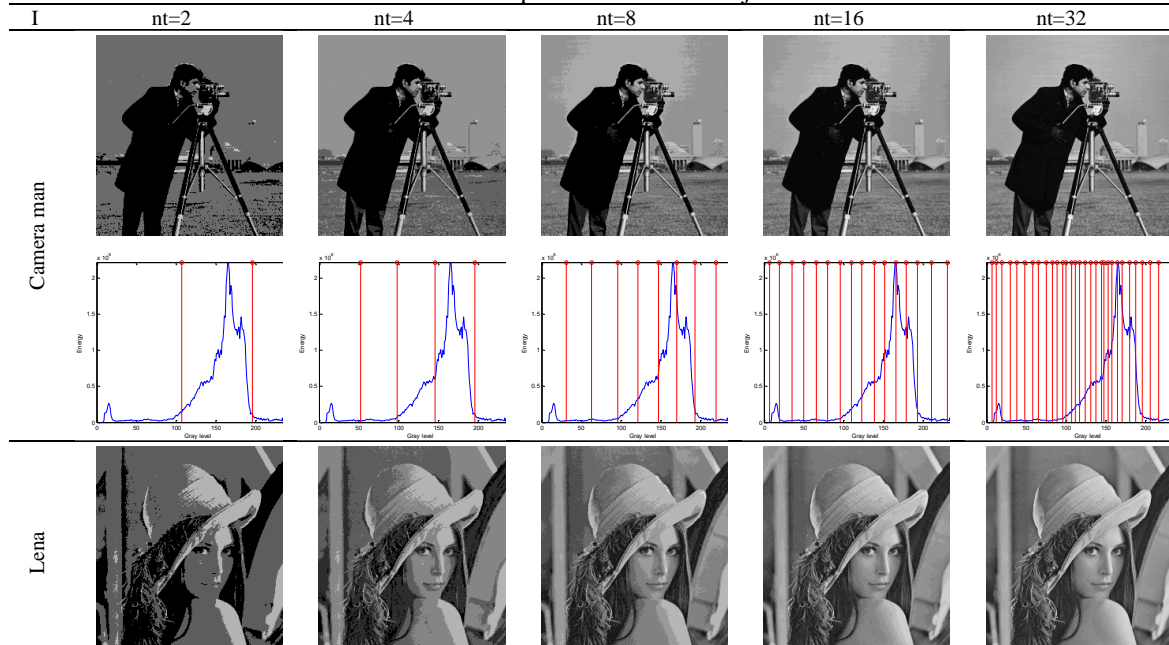
I	nt	ALO		SCA		GA		PSO		CSA		RRA		ACOR	
		Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ
Cameraman	2	0.7134	4.51E-16	0.7129	1.14E-03	0.7134	4.51E-16	0.7126	1.37E-03	0.5574	4.50E-16	0.7132	1.40E-03	0.6991	4.51E-16
	4	0.8318	1.96E-04	0.8277	1.13E-02	0.8312	2.54E-03	0.8284	1.99E-02	0.6324	6.80E-04	0.8326	2.80E-03	0.8151	1.97E-04
	8	0.883	1.58E-03	0.8713	2.09E-02	0.8842	3.42E-03	0.8687	1.75E-02	0.7343	1.60E-03	0.8834	3.80E-03	0.866	1.91E-03
	16	0.9288	2.04E-02	0.932	2.55E-02	0.9521	1.63E-02	0.9179	2.66E-02	0.8493	2.20E-02	0.9385	2.20E-02	0.9223	2.28E-02
	32	0.9773	1.44E-02	0.959	1.91E-02	0.9786	6.28E-03	0.9614	1.91E-02	0.948	8.40E-03	0.9788	1.20E-02	0.9676	1.31E-03
Lena	2	0.6255	2.06E-06	0.6259	2.40E-03	0.6255	0.00E+00	0.6262	3.67E-03	0.3927	0.00E+00	0.6255	1.50E-06	0.613	2.25E-16
	4	0.8118	1.63E-02	0.8086	1.52E-02	0.8151	3.51E-03	0.7738	3.78E-02	0.6074	8.70E-05	0.8146	7.70E-03	0.7952	1.33E-02
	8	0.9041	1.32E-02	0.887	1.64E-02	0.9074	4.06E-03	0.8758	3.11E-02	0.7992	3.20E-03	0.9035	7.50E-03	0.8897	2.03E-03
	16	0.9646	5.86E-03	0.935	1.48E-02	0.9626	6.67E-03	0.9337	1.73E-02	0.9061	3.90E-03	0.9635	5.20E-03	0.9479	1.72E-03
	32	0.9845	4.86E-03	0.9691	7.34E-03	0.9853	2.12E-03	0.9729	6.87E-03	0.963	1.40E-03	0.9875	2.60E-03	0.9694	8.30E-04
Hunter	2	0.6055	2.35E-07	0.6008	6.47E-03	0.6036	3.76E-03	0.6017	6.66E-03	0.3445	1.30E-07	0.6046	2.60E-03	0.5934	2.23E-07
	4	0.8367	1.71E-03	0.8289	1.57E-02	0.8378	6.89E-03	0.8367	1.36E-02	0.5043	1.90E-03	0.8353	6.90E-03	0.8212	1.78E-03
	8	0.9189	4.69E-03	0.9284	1.51E-02	0.9307	8.16E-03	0.9066	3.85E-02	0.692	7.60E-03	0.9201	9.70E-03	0.917	4.43E-03
	16	0.9733	1.89E-03	0.9636	1.21E-02	0.9786	3.51E-03	0.9542	1.62E-02	0.8435	4.10E-03	0.9733	6.20E-03	0.9611	1.34E-03
	32	0.993	1.44E-03	0.9835	5.09E-03	0.9935	1.37E-03	0.9825	5.49E-03	0.944	2.90E-03	0.9912	2.80E-03	0.9758	3.81E-04

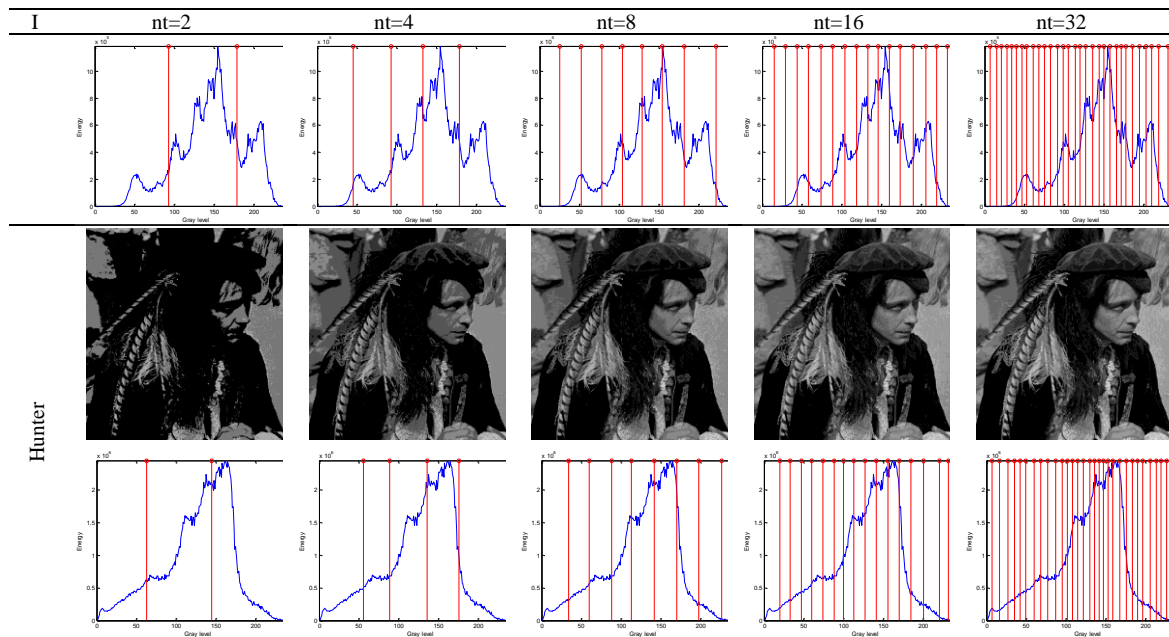
Tabla 5.10. Media y σ de los valores de FSIM obtenidos por los métodos SCA, ALO, GA, PSO, CSA, RRA y ACOR usando el método de Kapur.

I	nt	ALO		SCA		GA		PSO		CSA		RRA		ACOR	
		Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ	Media	σ
Cameraman	2	0.7229	4.50E-16	0.7215	1.10E-03	0.7229	4.50E-16	0.7208	1.40E-03	0.7063	5.60E-16	0.7234	3.50E-03	0.7084	5.63E-16
	4	0.837	2.00E-04	0.836	1.10E-02	0.8371	2.50E-03	0.8382	2.00E-02	0.7864	8.80E-04	0.8392	5.30E-03	0.8202	4.12E-04
	8	0.9147	1.60E-03	0.8829	2.10E-02	0.9138	3.40E-03	0.8982	1.70E-02	0.8974	2.70E-03	0.9137	4.50E-03	0.8963	3.42E-03
	16	0.9612	2.00E-02	0.9329	2.50E-02	0.9583	1.60E-02	0.9359	2.70E-02	0.953	4.10E-03	0.9583	5.60E-03	0.9411	1.93E-03
	32	0.983	1.40E-02	0.9631	1.90E-02	0.9814	6.30E-03	0.9653	1.90E-02	0.9834	2.40E-03	0.9836	3.00E-03	0.9676	9.03E-04
Lena	2	0.678	2.10E-06	0.678	2.40E-03	0.678	0.00E+00	0.6779	3.70E-03	0.6167	5.60E-16	0.678	5.30E-06	0.6644	5.63E-16
	4	0.758	1.60E-02	0.7605	1.50E-02	0.7581	3.50E-03	0.7629	3.80E-02	0.7224	6.00E-05	0.7588	5.20E-03	0.7417	4.85E-03
	8	0.8857	1.30E-02	0.848	1.60E-02	0.8853	4.10E-03	0.848	3.10E-02	0.857	4.50E-03	0.8822	9.70E-03	0.8674	3.26E-03
	16	0.9511	5.90E-03	0.904	1.50E-02	0.9483	6.70E-03	0.9067	1.70E-02	0.9422	5.50E-03	0.9491	5.50E-03	0.9356	2.94E-03
	32	0.9792	4.90E-03	0.9526	7.30E-03	0.9795	2.10E-03	0.9598	6.90E-03	0.9812	1.90E-03	0.9842	3.20E-03	0.9673	9.84E-04
Hunter	2	0.7099	2.30E-07	0.7069	6.50E-03	0.7086	3.80E-03	0.7075	6.70E-03	0.6677	1.00E-08	0.7094	1.70E-03	0.6957	1.80E-08
	4	0.8861	1.70E-03	0.8793	1.60E-02	0.8864	6.90E-03	0.8825	1.40E-02	0.816	8.90E-04	0.8856	3.10E-03	0.869	8.67E-04
	8	0.9501	4.70E-03	0.9433	1.50E-02	0.9543	8.20E-03	0.9343	3.80E-02	0.9363	4.50E-03	0.9493	3.90E-03	0.9393	3.92E-03
	16	0.987	1.90E-03	0.9685	1.20E-02	0.9891	3.50E-03	0.966	1.60E-02	0.983	2.40E-03	0.9866	3.50E-03	0.9709	7.09E-04
	32	0.9968	1.40E-03	0.9872	5.10E-03	0.9963	1.40E-03	0.9874	5.50E-03	0.9962	1.30E-03	0.996	1.50E-03	0.978	1.28E-04

Ahora, las imágenes segmentadas obtenidas por los métodos ALO y SCA usando Kapur se presentan para su inspección visual en las Tablas 5.11 y 5.12, respectivamente. Como en la sección previa, sólo se muestran tres imágenes representativas.

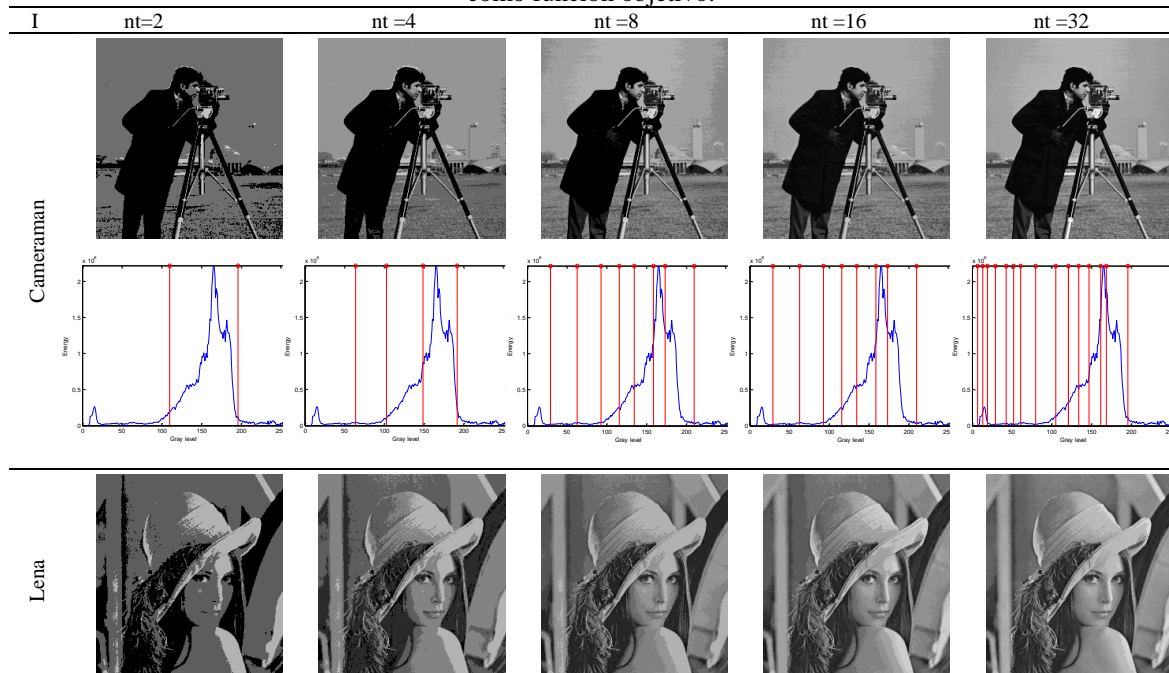
Tabla 5.11. Imágenes segmentadas y su curva de energía umbralada usando el algoritmo ALO y el método de Kapur como función objetivo.

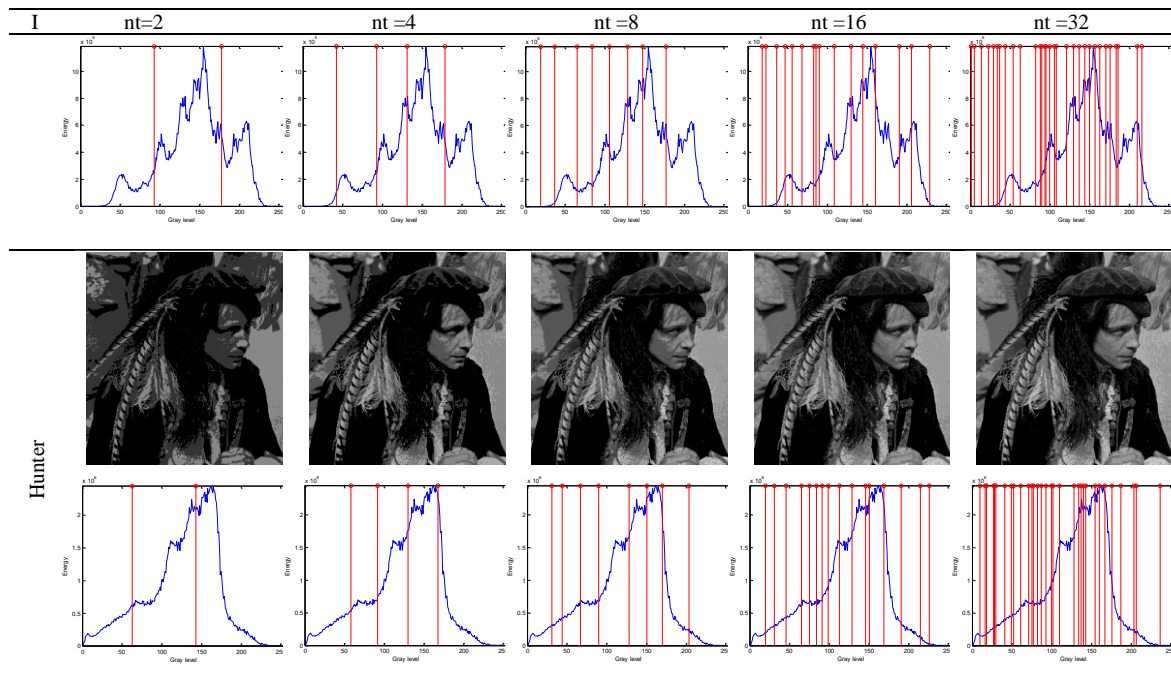




Por simple inspección de los umbrales que aparecen sobre las curvas de energía, puede fácilmente deducirse que las soluciones obtenidas usando el método de Kapur son diferentes a los umbrales generados por el método de Otsu a pesar de que usan el mismo algoritmo ALO en las mismas imágenes. Esta situación ocurre dado que cada función objetivo trabaja con información distinta extraída de la curva de energía (varianza entre clases y entropía). En la tabla 5.12 se muestran los resultados de SCA usando Kapur sobre las imágenes de prueba.

Tabla 5.12. Imágenes segmentadas y curva de energía umbralada por el algoritmo SCA usando Kapur como función objetivo.





A partir de la Tabla 5.12 se puede observar que las soluciones generadas por el algoritmo SCA no son buenas en todos los casos. Esta situación daña al umbralización porque sólo unos pocos valores de intensidad se eligen para generar clases pequeñas. De esos resultados es posible concluir que el algoritmo SCA falla en algunos casos para encontrar valores óptimos en espacios de muchas dimensiones. En la Figura 5.6 es posible comparar el tiempo promedio que cada algoritmo utiliza donde cada línea vertical indica medio segundo, donde el algoritmo ALO encuentra las soluciones en menor tiempo.

En esta subsección las Tablas 5.8-5.10 mostraron evidencia del desempeño de cada uno de los enfoques evaluados. A partir de tales resultados es posible concluir que el algoritmo ALO es el método evaluado que obtiene mejores resultados para la umbralización de imágenes digitales considerando ambas funciones objetivo definidas por Otsu y Kapur.

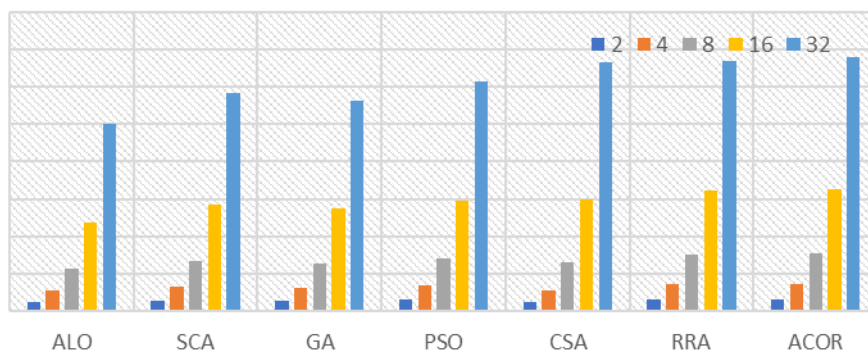


Figura 5.6. Tiempo promedio de cada algoritmo usando Kapur y 2, 4, 8, 16 y 32 umbrales.

5.3.3 Comparación estadística

En esta subsección se presenta un estudio comparativo entre los resultados obtenidos por los algoritmos ALO, SCA, GA, PSO, CSA, RRA y ACOR. Dado que todos estos algoritmos emplean números aleatorios, los resultados pueden variar de una evaluación a otra. Los valores de la función objetivo de cada método se comparan estadísticamente con la prueba no paramétrica de significancia de Wilcoxon, ya utilizada previamente y que, como en casos anteriores, se realiza con 35 muestras independientes. El análisis se lleva a cabo considerando un 5% de significancia sobre los valores de Otsu. Este test se evalúa sobre cada imagen considerando distintos números de umbrales nt . La Tabla 5.13 reporta los valores p producidos por las pruebas de Wilcoxon considerando comparaciones dos a dos entre ALO vs. GA, ALO vs. PSO, ALO vs. CSA, ALO vs. RRA, ALO vs. ACOR, SCA vs. GA, SCA vs. PSO, SCA vs. CSA, SCA vs. RRA, SCA vs. ACOR y ALO vs. SCA.

Tabla 5.13. Valores p de la prueba de Wilcoxon realizada sobre los valores de la función objetivo de Otsu.

I	nt	Valores p										
		ALO vs.					CSA vs.					
		GA	PSO	CSA	RRA	ACOR	SCA	GA	PSO	CSA	RRA	ACOR
Cameraman	2	8.16E-02	1.51E-08	4.57E-01	4.24E-02	5.85E-07	2.50E-07	8.01E-06	1.23E-01	2.50E-07	2.50E-07	2.50E-07
	4	1.53E-14	1.53E-14	3.81E-01	4.95E-08	5.32E-14	1.53E-14	6.89E-12	4.17E-01	2.18E-14	3.02E-14	3.17E-01
	8	5.71E-13	5.71E-13	2.00E-01	4.31E-03	5.70E-13	5.71E-13	6.54E-13	5.74E-08	6.48E-13	6.54E-13	6.53E-13
	16	6.54E-13	6.54E-13	2.00E-02	3.07E-02	1.22E-08	6.55E-13	6.54E-13	7.49E-12	6.55E-13	6.55E-13	6.55E-13
	32	1.53E-12	1.97E-12	3.78E-01	2.09E-01	4.33E-02	6.55E-13	6.54E-13	3.21E-11	6.55E-13	6.55E-13	6.55E-13
Lena	2	2.20E-05	3.49E-06	1.60E-01	5.69E-01	1.50E-06	1.85E-09	6.33E-07	2.92E-01	1.85E-09	1.85E-09	1.85E-09
	4	1.53E-14	1.53E-14	6.38E-08	1.19E-06	7.78E-06	1.53E-14	2.15E-11	1.34E-04	5.37E-14	2.18E-14	3.64E-13
	8	7.84E-08	1.77E-09	6.59E-01	1.24E-05	2.81E-07	6.38E-13	6.54E-13	2.08E-10	6.54E-13	6.55E-13	6.55E-13
	16	1.09E-12	4.56E-12	8.88E-01	1.71E-02	6.14E-01	6.55E-13	6.54E-13	4.57E-12	6.55E-13	6.55E-13	6.55E-13
	32	7.13E-13	2.77E-12	2.72E-02	8.15E-04	1.72E-07	6.55E-13	6.54E-13	1.12E-11	6.55E-13	6.55E-13	6.55E-13
Hunter	2	8.17E-02	1.49E-08	3.33E-02	1.78E-04	9.85E-06	5.85E-07	2.85E-05	7.09E-02	5.85E-07	5.85E-07	5.85E-07
	4	1.53E-14	1.53E-14	9.53E-01	2.85E-09	2.38E-07	1.53E-14	1.21E-10	4.03E-02	1.08E-13	2.48E-13	4.49E-13
	8	5.91E-13	5.91E-13	5.06E-04	6.98E-08	3.97E-12	5.92E-13	6.54E-13	9.25E-09	6.52E-13	6.54E-13	6.54E-13
	16	6.54E-13	6.54E-13	3.16E-08	1.44E-11	7.42E-01	6.55E-13	6.54E-13	2.96E-11	6.55E-13	6.55E-13	6.55E-13
	32	6.54E-13	6.54E-13	9.25E-04	8.21E-02	4.80E-10	6.55E-13	6.54E-13	1.61E-07	6.55E-13	6.55E-13	6.55E-13

Como hipótesis nula se asume que no existe diferencia entre los valores generados por los dos algoritmos en cuestión. La hipótesis alternativa considera la existencia de una diferencia entre los valores de ambos métodos. En la Tabla 5.13, se observa que los valores p para todas las comparaciones son menores a 0.05 (5%) lo cual es evidencia en contra de la hipótesis nula. Este hecho indica que los resultados obtenidos por ALO son significativamente distintos a los de cualquier otro algoritmo evaluado y no sucedieron por casualidad. Además, en la Tabla 5.14 se muestran los resultados de los valores p para las comparaciones entre ALO vs. GA, ALO vs. PSO, ALO vs. CSA, ALO vs. RRA, ALO vs. ACOR, SCA vs. GA, SCA vs. PSO, SCA vs. CSA, SCA vs. RRA, SCA vs. ACOR y ALO vs. SCA considerando los resultados de Kapur.

Tabla 5.14. Valores p de la prueba de Wilcoxon realizada sobre los valores de la función objetivo de Kapur.

I	nt	Valores p										
		ALO vs.					SCA vs.					
		GA	PSO	CSA	RRA	ACOR	SCA	GA	PSO	CSA	RRA	ACOR
Cameraman	2	3.32E-04	5.60E-09	4.24E-02	4.57E-01	1.77E-01	2.46E-12	2.46E-12	7.31E-02	5.30E-08	2.46E-12	2.46E-12
	4	5.89E-12	2.18E-14	4.95E-08	3.81E-01	3.58E-01	2.18E-14	9.99E-13	8.64E-02	1.47E-11	3.02E-14	3.02E-14
	8	6.30E-04	7.11E-13	4.31E-03	2.00E-01	2.73E-03	7.11E-13	6.55E-13	2.72E-02	8.47E-13	6.27E-13	6.49E-13
	16	5.17E-11	6.55E-13	3.07E-02	2.00E-02	5.81E-01	6.55E-13	6.55E-13	5.54E-04	6.55E-13	6.55E-13	6.55E-13
	32	3.01E-12	6.55E-13	2.09E-01	3.78E-01	1.18E-05	6.55E-13	6.55E-13	3.24E-01	6.55E-13	6.55E-13	6.55E-13
Lena	2	1.60E-01	6.36E-10	5.69E-01	1.60E-01	1.60E-01	1.59E-09	2.47E-10	4.97E-01	6.48E-10	2.47E-10	2.47E-10
	4	1.61E-09	1.83E-12	1.19E-06	6.38E-08	7.76E-04	5.98E-12	1.09E-12	1.30E-06	5.51E-11	2.18E-14	5.41E-11

	8	3.18E-11	1.28E-12	1.24E-05	6.59E-01	4.51E-08	9.94E-13	6.55E-13	5.26E-01	6.55E-13	6.50E-13	6.52E-13
	16	3.83E-10	6.55E-13	1.71E-02	8.88E-01	3.85E-01	6.55E-13	6.55E-13	1.12E-02	6.55E-13	6.55E-13	6.55E-13
	32	3.44E-07	6.55E-13	8.15E-04	2.72E-02	3.27E-12	6.55E-13	6.55E-13	2.49E-02	6.55E-13	6.55E-13	6.55E-13
Hunter	2	1.77E-04	2.67E-11	1.78E-04	3.34E-02	9.99E-02	2.56E-11	3.37E-06	2.57E-01	3.46E-04	2.56E-11	2.56E-11
	4	9.04E-12	2.13E-13	2.85E-09	9.53E-01	4.73E-01	2.13E-13	6.53E-13	4.88E-01	1.19E-12	1.55E-13	3.93E-13
	8	1.30E-12	6.54E-13	6.98E-08	5.06E-04	3.00E-12	6.54E-13	6.55E-13	4.66E-01	6.54E-13	6.47E-13	6.52E-13
	16	6.55E-13	6.55E-13	1.44E-11	3.16E-08	9.22E-13	6.55E-13	6.55E-13	1.00E-01	6.55E-13	6.55E-13	6.55E-13
	32	1.69E-11	6.55E-13	8.21E-02	9.25E-04	3.56E-05	6.55E-13	6.55E-13	1.39E-01	6.55E-13	6.55E-13	6.55E-13

Los valores mostrados en la Tabla 5.14 evidencian que el algoritmo ALO genera soluciones significativamente distintas al resto de métodos evaluados con la función objetivo de Kapur. Estos resultados junto con las métricas de calidad nos permiten afirmar que el algoritmo ALO es superior a los algoritmos evaluados para la umbralización de imágenes digitales.

5.4 Discusión

Los resultados obtenidos por los algoritmos propuestos (concretamente ALO) se comparan con métodos similares como se muestra en las Tablas 5.1-5.6 usando Otsu y las Tablas 5.7-5.11 usando Kapur. Tales tablas incluyen resultados gráficos que ilustran el proceso de segmentación realizado por los algoritmos ALO y SCA sobre la curva de energía. Los resultados de la Tabla 5.1 y la Tabla 5.7 indican que prácticamente todos los algoritmos tienen el mismo desempeño cuando $nt=2$. Esta situación ocurre ya que el problema de umbralización contiene sólo dos dimensiones. En contraste, cuando el número de umbrales a encontrar se incrementa, los algoritmos tienden a entregar soluciones subóptimas que afectan la calidad de las imágenes segmentadas. Considerando tales hechos, el algoritmo ALO es capaz de realizar mejores búsquedas en espacios de muchas dimensiones, en comparación con el algoritmo SCA.

Las Tablas 5.2-5.4 y las Tablas 5.8-5.10 muestran la calidad de los resultados de las imágenes segmentadas usando los umbrales obtenidos por los algoritmos seleccionados y usando como función objetivo Otsu y Kapur. Los valores de las métricas PSNR, SSIM y FSIM se incrementan de acuerdo con el número de umbrales. Aquí es importante mencionar que el uso de la curva de energía ayuda a tener más información de la distribución de los píxeles en la imagen (Ghosh et al., 2007; Patra et al., 2014) que cuando se utiliza el histograma. Los resultados logrados por el algoritmo ALO son mayores en la mayoría de los casos. Este comportamiento ocurre ya que cada imagen tiene distintas características que pueden dificultar o simplificar el problema de optimización. Además, la aleatoriedad de los algoritmos metaheurísticos genera variación en los resultados. Finalmente, basado en el teorema NFL (Wolpert & Macready, 1997) y usando los resultados presentados en las secciones previas, es posible concluir que es difícil definir si un algoritmo puede segmentar cualquier imagen correctamente. Esto también está soportado por el hecho de que la curva de energía es multimodal. Por ejemplo, si un umbral s es en una zona de la curva de energía que no es apropiada, la segmentación no será la mejor. Tal situación solo podrá verse reflejada a través de métricas

especializadas como PSNR, SSIM y FSIM porque la función objetivo tanto la de Otsu como la de Kapur trabaja con información limitada.

5.5 Resumen

En este capítulo se presenta un estudio sobre técnicas aplicadas a la umbralización multinivel de imágenes incorporando la reciente formulación de la curva de energía como un concepto nuevo y que permite añadir información contextual existente en la vecindad de cada píxel de la imagen, a diferencia del histograma que sólo contiene información de los niveles de intensidad. La evaluación se realiza considerando dos criterios clásicos de umbralización (Otsu y Kapur) donde estos criterios se aplican sobre la curva de energía en lugar de realizarlo sobre el histograma como en sus definiciones originales. El desempeño de tal formulación se evalúa exhaustivamente usando ocho algoritmos evolutivos donde el algoritmo ALO obtiene el mejor resultado.

6 Segmentación de imágenes con un enfoque multiobjetivo

Las técnicas tradicionales de umbralización se basan únicamente en la información presente en el histograma de la imagen, centrándose principalmente en dos familias de métodos: enfoques paramétricos y no paramétricos (Akay & Bahriye, 2013). El primer enfoque trata de aproximar cada clase como una función de densidad de probabilidad; sin embargo, resulta costoso desde el punto de vista computacional. Por otro lado, el enfoque no paramétrico propone umbrales candidatos y posteriormente evalúa la calidad de la segmentación generada mediante criterios no paramétricos como la varianza entre clases, la entropía y la divergencia, entre otros (Kapur et al., 1985; Li & Lee, 1993; Otsu, 1979). El procedimiento de umbralización no paramétrico puede dividirse en dos partes: en primer lugar, un criterio no paramétrico para evaluar la calidad de una imagen segmentada a partir de un determinado conjunto de umbrales, y, en segundo lugar, un método de búsqueda (por ejemplo, un algoritmo de optimización) que propone el conjunto de umbrales candidatos. Ambas partes trabajan juntas para mejorar iterativamente la calidad de los valores de umbral guiados por el criterio no paramétrico hasta que el proceso alcanza una solución óptima.

La mayoría de los enfoques basados en umbrales y clústeres requieren que el número de niveles o clases de intensidad se establezca *a priori*. Estos métodos son adecuados para aplicaciones en las que las imágenes procesadas comparten propiedades comunes y sus histogramas no varían significativamente, como la detección celular, las imágenes satelitales y las imágenes médicas (Nawandhar, Yamujala, & Kumar, 2015; Oliva, Hinojosa, Cuevas, et al., 2017; Sarkar, Das, & Chaudhuri, 2016). En este sentido, estas técnicas requieren la asistencia de

un diseñador para seleccionar cuántos umbrales representan con precisión la imagen original. La selección automática de ambos, el número y la colocación de los umbrales contribuyen a una técnica de umbralización general no asistida capaz de segmentar imágenes sin conocimiento previo o intervención externa.

Sin embargo, sólo unas pocas técnicas de umbralización tienen en cuenta la determinación automática del número de umbrales, centrándose en la búsqueda exhaustiva de la mejor ubicación de los umbrales y aumentando gradualmente el número de umbrales. Djerou et al. (2012) propuso un enfoque de umbralización no supervisado basado en la optimización multiobjetivo (MO) y un PSO binario donde el objetivo es encontrar el número óptimo de umbrales y buscar simultáneamente el conjunto óptimo de umbrales. Más tarde, Zheng (2016) publicó un método de segmentación adaptativa con umbralización multinivel basada en un algoritmo evolutivo multiobjetivo (MOEA) en el que se propone un cruce híbrido, y el número de umbrales se selecciona después de un análisis de error cuadrático. Sin embargo, ambos métodos presentan serios inconvenientes; la población se divide en un número fijo de subpoblaciones donde los elementos de cada subpoblación tienen el mismo número de umbrales. Este mecanismo está más cerca de buscar de forma independiente los mejores umbrales de una determinada población y luego aumentar exhaustivamente el número de umbrales. Además, como el tamaño de la subpoblación es fijo, la búsqueda está limitada por el número de agentes de búsqueda presentes en la población del número óptimo de umbrales, mientras que las partículas en el número no óptimo de poblaciones de umbrales no contribuyen directamente al resultado de la búsqueda.

Por definición, los algoritmos evolutivos de optimización multiobjetivo (MOEAs) tratan problemas que requieren encontrar soluciones óptimas para dos o más objetivos. Contrariamente a los algoritmos evolutivos de objetivo único (EAs), los MOEAs producen como resultado un conjunto de soluciones donde cada solución representa un equilibrio óptimo entre los objetivos (Mirjalili, Jangir, Mirjalili, Saremi, & Trivedi, 2017). La idea de la optimización estocástica multiobjetivo fue introducida por David Schafer (Rovcanin et al., 2014) con la incorporación de metodologías evolutivas que proporcionan características tales como la evasión de óptimos locales y búsquedas sin gradientes. Algunos de los métodos más relevantes incluyen, entre otros, el Algoritmo Genético de Clasificación No Dominado (NSGA) (Kalyanmoy Deb, 1999), NSGA-II (K Deb, Pratab, Agarwal, & Meyarivan, 2002), el Algoritmo Evolutivo de Pareto de Fuerza (SPEA) (Eckart Zitzler & Thiele, 1999), SPEA-2 (E Zitzler, Laumanns, & Thiele, 2001), la Optimización de Enjambres de Partículas Multiobjetivo (MOPSO) (Coello et al., 2004) y el más recientemente propuesto NSGA-III (Kalyanmoy Deb & Jain, 2013). Estos métodos pueden manejar optimización continua y combinatoria con alta precisión y velocidad de convergencia aceptable (Suman, 2005). Los MOEAs optimizan

simultáneamente dos o más objetivos a la vez que consideran la existencia de una relación entre los objetivos a través de la compensación entre los objetivos actuales.

Las cuatro contribuciones principales de este capítulo pueden resumirse como sigue:

1. La propuesta de una metodología genérica de Umbralización No Asistido (UTH) (Hinojosa, Avalos, et al., 2018) basada en MOEAs para realizar umbralización de imagen mientras se determina simultáneamente el número óptimo de umbrales y su mejor ubicación.
2. La evaluación de UTH usando tres MOEAs (Hinojosa, Avalos, et al., 2018).
3. El uso de partículas de longitud variable en el proceso evolutivo multiobjetivo (Hinojosa, Avalos, et al., 2018).
4. Propuesta de un nuevo algoritmo evolutivo multiobjetivo (Hinojosa et al., 2017).

La primera contribución se formula como la optimización de dos objetivos contradictorios; un objetivo pretende mejorar la calidad de la imagen segmentada mediante la minimización de la entropía cruzada, mientras que el otro objetivo se dedica a minimizar el número de umbrales utilizados para la segmentación ya que la calidad de una imagen segmentada aumenta a medida que se incrementa el número de umbrales. Teóricamente, la máxima calidad de una imagen segmentada se alcanza cuando el número de umbrales se acerca al número original de niveles de gris, es decir, cuando cada clase corresponde a un único valor de intensidad. Si bien esto es cierto, el establecimiento de umbrales con un número elevado de umbrales no tiene sentido. Por lo tanto, debe especificarse un límite sobre cuántos umbrales se utilizan. En este contexto, los MOEAs aplicados a la metodología UTH generan como salida un conjunto de partículas con un equilibrio óptimo entre la calidad de imagen y el número de umbrales. Después de esto, se selecciona la partícula más cercana al origen de la salida normalizada para realizar la segmentación (Kasprzak & Lewis, 2000).

La segunda contribución se centra en la evaluación del desempeño de la metodología genérica de UTH en un conjunto con MOEAs. David Wolpert en una serie de pruebas lógicas propone el teorema *No-Free-Lunch* para la optimización (Wolpert & Macready, 1997), donde afirma que ningún algoritmo de optimización funcionará bien en todos y cada uno de los problemas. Por lo tanto, la evaluación de un determinado MOEA sobre una tarea específica no podía reflejar con precisión las capacidades del método propuesto. Para evaluar el desempeño de la UTH, se integran tres MOEAs representativas con la metodología propuesta, NSGA-III, PESA-2 y MOPSO. Tanto el MOPSO como el PESA-2 han sido seleccionados por su relevancia sobre los MOEAs y el NSGA-III se incluye por su relativa novedad. Por lo tanto, los métodos acoplados serán referidos como UTH-NSGA-III, UTH-PESA-II, y UTH-MOPSO

respectivamente. Dado que NSGA-III es el más reciente de los tres, se presta especial atención a UTH-NSGA-III.

La tercera contribución en el enfoque propuesto es el uso de partículas de longitud variable en el proceso evolutivo. Los enfoques de umbralización tradicionales codifican en cada partícula del EA un conjunto de umbrales para segmentar la imagen. Dado que el número de umbrales es definido *a priori* por el diseñador, todas las partículas tienen la misma longitud. En el contexto de la UTH, las partículas con diferentes longitudes existen en la misma población e interactúan entre sí. Esto se consigue a través de un mecanismo en el que cada partícula puede mantener un número máximo de umbrales, y reducir o aumentar dinámicamente el número de umbrales si es necesario.

Finalmente, las tres implementaciones de UTH se comparan con métodos relacionados y luego se evalúa la calidad de las imágenes segmentadas y la convergencia de los algoritmos utilizando un conjunto de imágenes de prueba ampliamente conocidas en las comunidades de visión por computador y procesamiento de imágenes. La convergencia se evalúa en función del hipervolumen y de las superficies de logro generadas por cada implementación descritas en la subsección 6.2.5.1. Los resultados proporcionados por la experimentación soportan UTH-NSGA-III como una buena alternativa para realizar automáticamente la umbralización de imágenes sin la configuración o supervisión de un diseñador. Aún más, este trabajo fomenta el uso de NSGA-III, ya que supera a MOPSO y PESA-2 para esta aplicación basada en imágenes.

6.1 Descripción de algoritmos multiobjetivo

La principal ventaja de la metodología UTH propuesta sobre las técnicas de umbralización tradicionales radica en su naturaleza no asistida. La mayoría de las técnicas de umbrales requieren que el número de umbrales se determine *a priori* con la ayuda de un experto, mientras que UTH determina tanto el número de umbrales como la colocación óptima de tales valores a lo largo del histograma de la imagen. Además, los resultados indican que el rendimiento de las metodologías de la UTH es superior al de los métodos relacionados. En esta sección se describe brevemente cada uno de los algoritmos usados: NSGA-III, PESA-II, y MOPSO.

6.1.1 NSGA-III

El algoritmo NSGA-III fue presentado por Deb (Kalyanmoy Deb & Jain, 2013), el cual es una mejora del algoritmo NSGA-II (Kalyanmoy Deb et al., 2002) con diferencias importantes en el operador de selección. NSGA-III usa una población de N individuos inicializados aleatoriamente P_t , después, se crea una población de descendientes Q_t e a partir de P_t y tamaño N , a continuación, la población S_t se ordena a partir de la combinación de las otras dos

poblaciones $S_t = P_t \cup Q_t$ usando los niveles de no-dominancia de las funciones objetivo. El nivel de no-dominación (l) se selecciona de una nueva población R_t hasta que el número de N individuos es alcanzado. Para determinar los niveles de no-dominancia, todos los individuos de la población usan el principio básico de la dominación, si $|R_t| = N$ la siguiente generación se inicia con $P_{t+1} = R_t$. Para $|R_t| > N$ los individuos de 1 a $l-1$ son seleccionados del último frente. Para asegurar la diversidad de la población, el algoritmo NSGA-III predefine un conjunto de puntos de referencias conocidos como es el enfoque sistemático de Das y Dennis, el cual usa los puntos de un hiperplano normalizado. El punto de referencia H está dado por la siguiente ecuación,

$$H = \binom{M+p-1}{p}, \quad (6.1)$$

Otra técnica utilizada por NSGA III es el uso de una población normalizada, la cual se denota como $f'_i(\mathbf{x}) = f_i(\mathbf{x}) - z_i^{\min}$ y se obtiene restando el objetivo f_i con el valor mínimo z_i^{\min} , así, estos vectores se usan para formar un hiperplano. La normalización de la función objetivo se establece como,

$$f_i^n(\mathbf{x}) = \frac{f'_i}{a_i}, \quad (6.2)$$

donde a_i es la intersección del eje del i -ésimo objetivo y el hiperplano. Después de la normalización, las funciones objetivo se asocian a cada elemento de la población con un punto de referencia. Usando el origen se genera una línea hasta cada punto de referencia para calcular la distancia perpendicular entre cada elemento de R_t y cada línea de referencia. En este caso, el punto de referencia se considera como un elemento de la población si su línea de referencia es la más cercana a un elemento de la población en el espacio de objetivos normalizados. El número de elementos de la población asociados a cada línea de referencia puede ser uno o más. Es por ello por lo que se usa el operador *niche-preserving* (Kalyanmoy Deb & Jain, 2013), estableciendo ciertas restricciones para determinar si las soluciones deben ser consideradas, si existe más de un elemento, uno de ellos deberá ser seleccionado aleatoriamente. Los operadores de cruce para generar nuevos descendientes son los mismos que los establecidos por NSGA-II (Kalyanmoy Deb et al., 2002).

6.1.2 PESA-II

El algoritmo PESA-II fue introducido por Corne et al. (2001) como una mejora del algoritmo de selección basada en la envolvente de Pareto Corne et al. (2000). PESA-II utiliza los mecanismos del algoritmo genético tradicional con un proceso basado en regiones donde se extraen celdas, pero no soluciones individuales en el proceso de selección. Con este mecanismo,

una celda muy ocupada tiene menos posibilidades de ser seleccionada que una celda poco ocupada. Después de seleccionar una celda, las soluciones de esa celda se eligen al azar para que sean padres o mutantes. Además, PESA-II utiliza una memoria externa llamada archivo para almacenar las soluciones aproximadas de Pareto.

6.1.3 MOPSO

El algoritmo MOPSO introducido por Coello et al. (2004) genera aleatoriamente una población de N partículas a lo largo del espacio de búsqueda. Cada elemento de la población es evaluado para determinar si está dominado por otras partículas. Los elementos no dominados se almacenan en una memoria externa denominada archivo. Los siguientes operadores son los mismos que los usados por PSO. La posición y la velocidad de cada partícula en la población se calculan en cada generación considerando un peso de aceleración variable en el tiempo. En MOPSO la mejor solución global es una partícula tomada del archivo. Si se determina que una solución es inviable, la partícula toma un valor límite. La posición del mejor local se actualiza sólo si la posición actual domina el mejor local anterior. En el caso de dos soluciones que no se dominan mutuamente se selecciona aleatoriamente una de las dos. Al final de cada generación, el archivo se actualiza para almacenar las soluciones no dominadas actuales.

6.2 Umbralización sin asistencia (UTH)

En esta sección, se describe la metodología de umbralización de imágenes sin asistencia, *Unassisted Thresholding* (UTH) basada en MOEAs (Hinojosa, Avalos, et al., 2018). UTH trata con los dos problemas principales de la umbralización de imágenes; la determinación del número de umbrales, y la colocación óptima de los umbrales sobre el histograma de la imagen. UTH aprovecha los mecanismos de compromiso presentes en los MOEAs para realizar la umbralización de imagen mientras que simultáneamente determina el número de umbrales requeridos para segmentar cada imagen y la mejor colocación de cada umbral a lo largo del histograma de la imagen sin ningún conocimiento previo o la intervención de un diseñador o supervisor.

La calidad de una imagen segmentada está muy relacionada con el número de umbrales que intervienen en el proceso. En teoría, el mejor escenario de la umbralización ocurre cuando el número de clases segmentadas es igual al número de niveles de intensidad. Así, la imagen segmentada sería igual a la imagen original. Sin embargo, este escenario no tiene sentido, ya que no implica proceso de segmentación alguno. Como resultado, las técnicas de umbralización deben establecer un límite sobre cuántos valores umbral se seleccionan para una imagen dada. Con esto en mente, es fácil ver que los MOEAs pueden aplicarse a la metodología UTH para buscar un conjunto de partículas con un equilibrio óptimo entre la calidad de la imagen y el número de umbrales. De este conjunto de soluciones óptimas, se utiliza un simple proceso de

toma de decisiones para decidir qué partícula del conjunto de salida se selecciona para realizar la segmentación.

En la metodología UTH propuesta, el problema de la umbralización se considera una tarea de optimización con dos objetivos. El primer objetivo está dedicado a la búsqueda de valores umbral óptimos capaces de aumentar la calidad de la imagen segmentada. El segundo objetivo se refiere a la minimización del número de umbrales. La combinación de estos dos objetivos es conflictiva. El primer objetivo se beneficia del detrimento del segundo. Dado que el número de umbrales puede tener un gran impacto en la calidad del resultado, un gran número de umbrales beneficia al resultado del proceso de segmentación, mientras que un pequeño número de umbrales genera resultados de baja calidad. Los MOEAs generan intrínsecamente un equilibrio entre ambos objetivos. UTH utiliza estos mecanismos para proporcionar en la misma ejecución el número de umbrales y los valores de umbral óptimos.

En las siguientes subsecciones se detallará la definición de cada función objetivo involucrada en el proceso de segmentación. Después se presenta el funcionamiento de una partícula de longitud variable, la cual ayuda a aumentar la diversidad entre las soluciones de la población. Además, en este capítulo se detalla la operación de un operador de cruce mejorado para el algoritmo UTH-NSGA-III seguido por la explicación del mecanismo usado para elegir sólo una solución de entre todo el frente de Pareto encontrado y finalmente segmentar la imagen.

6.2.1 Funciones objetivo

En UTH, la primera función objetiva está dedicada a maximizar la calidad de la imagen segmentada utilizando un criterio no paramétrico. En este trabajo se evalúa un criterio representativo no paramétrico: la Entropía Cruzada (Ec. 2.28). Debe tenerse en cuenta que sólo se utiliza un criterio en este capítulo, pero la formulación de UTH no se limita a este criterio. La Entropía Cruzada se utiliza para evaluar la calidad de un determinado conjunto de umbrales \mathbf{th}

$$\begin{aligned} \arg \min_{\mathbf{th}} \quad & f_c(\mathbf{th}) \\ \text{sujeto a} \quad & \mathbf{th} \in \mathbf{X} \end{aligned} \tag{6.3}$$

con el conjunto de restricciones \mathbf{X} de tal manera que:

donde $f_c(\mathbf{th})$ es la función objetivo de la entropía cruzada (Ec.2.28) y $\mathbf{X} = \left\{ \mathbf{th} \in \mathbb{N}^{nt} \mid 0 \leq th_i \leq 255, i=1,2,\dots,nt \right\}$ contiene las restricciones de la región factible del espacio de búsqueda acotada por los posibles valores de intensidad de la imagen variando en el rango 0 a 255.

El segundo objetivo consiste en realizar la segmentación con el menor número de umbrales posible. En este contexto, la segunda función del objetivo es la minimización del número de valores utilizados para el umbral (nt). Para facilitar el proceso de toma de decisiones, la segunda función objetivo se define como el número de umbrales normalizado, de la siguiente manera:

$$\begin{aligned} \arg \min_{th} \quad & \frac{nt_i}{ntMax} \\ \text{sujeto a} \quad & ntMax > 1 \end{aligned} \quad (6.4)$$

6.2.2 Partícula de longitud variable

Cada umbral th es incorporado dentro del algoritmo de optimización como una variable de decisión en cada elemento de la población. Así, la población se representa como:

$$\mathbf{Sp}_t = \{\mathbf{th}_1, \mathbf{th}_2, \dots, \mathbf{th}_N\}, \mathbf{th}_i = [th_1, th_2, \dots, th_{nt_i}]^T \quad (6.5)$$

donde t es el número de iteración, N el tamaño de la población, T se refiere al operador de traspuesta y $nt_i \in \mathbb{N}^{ntMax}$ es el número de umbrales contenidos dentro de la partícula i . La población \mathbf{Sp} contiene partículas de tamaño máximo $ntMax$.

La incorporación de partículas de longitud variable en el proceso de optimización es una de las principales aportaciones de este planteamiento, recogido en (Hinojosa, Avalos, et al., 2018). Hasta donde se ha podido determinar el tamaño dinámico de la partícula no ha sido tratado previamente por la comunidad científica, y el enfoque más cercano podría considerarse el uso de subpoblaciones donde cada subpoblación tiene un tamaño fijo para las partículas contenidas, pero no hay interacción entre dos o más subpoblaciones.

Debido a que las partículas varían en tamaño, se deben hacer algunas consideraciones. En UTH, el proceso de inicialización de la MOEA seleccionada se modifica ligeramente para favorecer la diversidad en las soluciones. Primero, el tamaño nt_i de cada partícula \mathbf{th}_i de la población inicial \mathbf{Sp}_1 se inicializa aleatoriamente $[1, ntMax]$. Después, los valores de umbral de \mathbf{th}_i se generan siguiendo las reglas específicas del MOEA considerando el tamaño de cada partícula como nt_i . Este proceso permite generar una población donde el tamaño de las partículas siga una distribución uniforme. La figura 6.1 muestra cómo se usan umbrales inoperantes con valor igual a uno para completar el tamaño $ntMax$.

57	98	178				1	1	1	57	98	178
29	57	98	178			1	1	29	57	98	178
11	29	57	98	178		1	11	29	57	98	178

Partículas de longitud variable Partículas de longitud $ntMax$ con umbrales inoperantes

Figura 6.1. Representación de partículas en UTH.

6.2.3 Operador de cruce modificado

Los métodos de optimización codifican en cada partícula un vector de variables de decisión. Los operadores tradicionales presentes en los algoritmos evolutivos y basados en enjambres requieren una relación directa entre las variables de decisión. Más importante aún, la mayoría de los operadores evolutivos requieren dos o más partículas del mismo tamaño para generar una nueva partícula. En la UTH, este no es el caso, ya que una partícula que contiene tres umbrales no tiene una relación directa con una partícula que posee diez umbrales. Un primer intento de solucionar este problema podría ser la incorporación de umbrales ficticios. Por ejemplo, la partícula más pequeña podría acomodar un número dado de valores de umbral repetidos al principio de la partícula para alcanzar el mismo tamaño y permitir la aplicación de cualquier operador. Estos valores ficticios no perjudicarían la calidad de la segmentación, ya que se solapan. La figura 6.1 muestra ambas implementaciones prácticas, utilizando umbrales ficticios y partículas de tamaño dinámico. Con este enfoque, los operadores tradicionales aplicados a partículas con diferentes longitudes generarán nuevas soluciones con el tamaño de la partícula más grande. Si este proceso se itera a través de muchas generaciones, eventualmente todas las partículas tendrán umbrales $ntMax$. Para prevenir este fenómeno, se propone un operador de cruce modificado para los MOEAs donde las variables de decisión no son operadas considerando su posición en el vector sino más bien por su cercanía a sus valores.

El operador de cruce propuesto basado en la cercanía está diseñado para operar sólo para el problema del umbral. Se necesitan dos partículas típicamente conocidas como padres α y β para producir dos hijos denotados como α' y β' . La Ec. 6.6 muestra la operación del operador de cruce modificado,

$$\begin{aligned}\alpha'_i &= r \cdot \alpha_i + (1-r) \cdot \beta_c \\ \beta'_i &= r \cdot \beta_i + (1-r) \cdot \alpha_c\end{aligned}\tag{6.6}$$

donde β_c es el valor de umbral más cercano contenido en el padre β al valor de umbral α_i representa la mínima diferencia absoluta ($c \in \mathbb{N}^{m_\beta} \mid \min_c |\alpha_i - \beta_c|$). De forma similar, α_c es el valor de umbral más cercano contenido en el padre α al valor de umbral β_i tal que minimiza la

diferencia absoluta ($c \in \mathbb{N}^{m_a} \mid \min_c |\beta_i - \alpha_c|$). Además, r es un número aleatorio extraído de una distribución uniforme entre $[0,1]$. Con este operador de cruce se pueden trabajar partículas de diferentes tamaños. Seleccionando los valores de umbral más cercanos, se refuerzan los umbrales similares. La Figura 6.2 presenta gráficamente un ejemplo de este operador de cruce. En este ejemplo, se supone que el valor de r es 0,5 para simplificar. Cada flecha apunta al valor umbral más cercano seleccionado para el operador de cruce. Este mecanismo contribuye a diversificar el número de umbrales seleccionados. Como resultado, esta modificación evita un sesgo hacia soluciones con un gran número de umbrales.

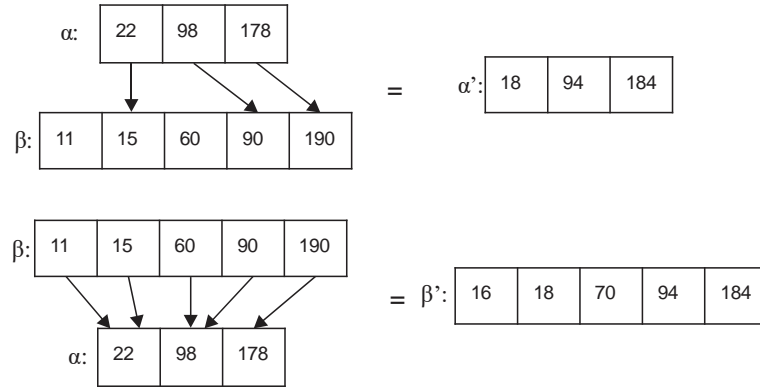


Figura 6.2. Operador de cruce modificado basado en la cercanía del valor.

6.2.4 Proceso de toma de decisiones y segmentación de imagen

El problema de los umbrales requiere una única salida en lugar de un conjunto de soluciones no dominadas. Un método para seleccionar una sola solución del conjunto de soluciones no dominadas es escoger la solución que minimice la distancia a un punto ideal, en este caso, el origen. Esto es posible porque el frente de Pareto de este problema es convexo y las dos funciones objetivas minimizan cada meta. Para ello, cada objetivo se normaliza para colaborar en la selección de la solución de salida por igual, y se selecciona la más cercana al origen. Este enfoque ha demostrado su eficacia desde su publicación (Kalyanmoy Deb & Sundar, 2006).

Una vez que el UTH junto con un MOEA deja de iterar, el proceso de toma de decisiones selecciona una única partícula de la salida para realizar la segmentación de la imagen mediante la regla de la Ec. 2.6.

Dado que la UTH es una metodología genérica, su incorporación a una MOEA específica es sencilla. Se resumen las modificaciones necesarias para implementar UTH. Cada partícula de la población debe ser implementada dinámicamente para aumentar o reducir su tamaño durante el proceso evolutivo. Una alternativa a este enfoque es definir partículas con un tamaño $ntMax$ y llenar las primeras posiciones $ntMax - nt_i$ con umbrales ficticios. Esto es

especialmente importante si el MOEA seleccionado no tiene un operador de cruce. Después de estas modificaciones, cualquier MOEA debería ser capaz de realizar UTH siguiendo los valores de cada función objetivo y los operadores de la MOEA seleccionada para guiar el proceso de evolución mientras se mejora la imagen segmentada.

6.2.5 Resultados experimentales

La metodología UTH propuesta se analiza en dos partes. La primera se dedica a comparar el rendimiento del método UTH con enfoques similares como el ATMO (Djerou et al., 2012) y AMT-ME (Zheng et al., 2016) en las mismas condiciones. Posteriormente, la segunda parte evalúa la metodología UTH propuesta utilizando tres MOEAs, NSGA-III, MOPSO y PESA-II, para identificar tanto el número óptimo de umbrales como su correcta ubicación en el histograma para un conjunto de imágenes ampliamente utilizadas en la literatura sobre visión por computador. La evaluación se realiza teniendo en cuenta la convergencia y la calidad de las imágenes segmentadas. La primera subsección describe las métricas utilizadas para evaluar el desempeño de cada método. La segunda subsección muestra el montaje experimental. A continuación, se presenta la comparación de UTH, ATMO y AMT-ME. Luego, en la subsección tres se presenta un análisis de convergencia de las metodologías de la UTH. Posteriormente, la calidad de las imágenes segmentadas se evalúa en la quinta subsección. Finalmente, la subsección seis discute los resultados generales y sus implicaciones.

6.2.5.1 Métricas

El desempeño de las MOEAs también se verifica usando métricas como el hipervolumen diseñado por Zitzler (Eckart Zitzler & Thiele, 1999). Tal métrica cuantifica el comportamiento de convergencia de los algoritmos multiobjetivo. El principio subyacente es el cálculo del área, volumen o hipervolumen del espacio objetivo dominado por las soluciones de Pareto obtenidas. Para que esto funcione, se fija un punto de referencia en el espacio de soluciones, y se calcula el hipervolumen existente entre el frente de Pareto y el punto de referencia. La elección del punto de referencia considera los valores máximos posibles de las soluciones presentes en el Frente de Pareto (en el caso de minimización). Debido a la naturaleza de las funciones de prueba, el punto fijo se establece como (3, 1) para el problema bi-objetivo en todos los algoritmos, ya que el valor máximo observado para el primer objetivo es de alrededor de 2,8 y se deja un poco de margen hasta 3 mientras que el segundo objetivo se normaliza, por lo que su valor máximo posible es uno.

Los valores generados por esta métrica son de 0 a 1 donde 1 indica el máximo hipervolumen posible desde el punto dado hacia el origen del espacio de búsqueda. Un valor más alto indica un mejor rendimiento.,

Por último, Fonseca y Flemming (Fonseca & Fleming, 1996) introdujeron el término superficie de logro, que es un límite que separa el espacio de objetivos en dos regiones: aquellas soluciones objetivas que son alcanzadas (o dominadas por o iguales a) los resultados generados por los algoritmos analizados. Los gráficos de superficie de logro (López-Ibáñez, Paquete, & Stützle, 2010) indican la mejor, la mediana y la peor superficie de logro. Se espera que los tres estén lo más cerca posible del verdadero Pareto.

6.2.5.2 *Diseño experimental*

Dado que la metodología UTH propuesta se analiza en dos partes, el diseño experimental de la primera parte replica las condiciones de Zheng et al. (2016) donde un conjunto de cinco imágenes sintéticas y naturales (Figura 6.3) se segmentan y se analiza el valor PSNR (Omran, Salman, & Engelbrecht, 2006). El número real de umbrales se determina visualmente según Zheng et al. (2016). Dado que tanto ATMO como AMT-ME utilizan subpoblaciones que buscan una serie de umbrales en lugar de una única población que contiene soluciones de todos los umbrales posibles como la UTH, no es posible realizar una comparación directa sobre la convergencia o la precisión. Así, en la segunda parte se analiza el desempeño de cada implementación de UTH. Para la prueba experimental, cada MOEA fue acoplada con la entropía cruzada en el primer objetivo mientras que el número normalizado de umbrales se selecciona como la segunda función del objetivo. Cada experimento se reporta tras 36 ejecuciones individuales en cada imagen del conjunto. Todos los experimentos se realizaron usando Matlab 8.3 (MathWorks, 2018) en un procesador con una CPU i5-4210 a 2.3Ghz con 6GB de RAM.

Los parámetros utilizados para los experimentos son los mismos de sus referencias a menos que se indique explícitamente para el NSGA III (Kalyanmoy Deb & Jain, 2013), en el caso del MOPSO (Parsopoulos, 2002), y para el PESA II (D. Corne et al., 2001). El tamaño de la población se fijó en 50 para NSGA-III, MOPSO y PESA-II. El tamaño de archivo para PESA-II y MOPSO es de 100. El número de iteraciones está limitado por la condición de parada para realizar hasta 25000 evaluaciones de función de aptitud de cada algoritmo (NSGA-III y PESA-II alrededor de 500 iteraciones mientras que 250 forman MOPSO). La probabilidad de cruzamiento de NSGA-III y PESA-II se establece en 0,7. El mecanismo de cuadrícula utilizado en MOPSO y PESA-II utiliza diez cuadrículas por dimensión y una tasa de inflación de 0,1. La probabilidad de mutación para ambos algoritmos es inversamente proporcional a la longitud del cromosoma.

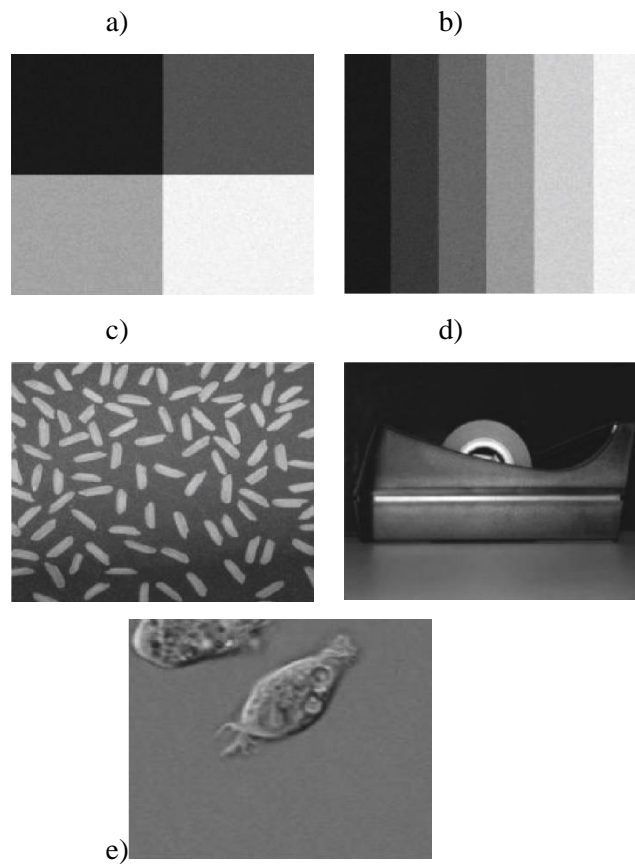


Figura 6.3. Primer conjunto de imágenes. a)Mmi3, b)Mmi5, c)Rice, d) Tape y,e) Cell.

6.2.5.3 Comparación de enfoques anteriores

En esta subsección, el desempeño de las tres implementaciones de UTH se compara con los resultados reportados por Zheng et al. (2016). Con este fin, se adopta el mismo proceso de evaluación que en la citada referencia para asegurar una comparación justa, donde se utilizan las mismas imágenes con poblaciones similares. El objetivo de la primera prueba es determinar si el mejor conjunto de umbrales para segmentar una imagen dada y comparar el número de umbrales con el óptimo. En la Tabla 6.1 se muestran los resultados, observándose cómo todos los métodos logran encontrar el número óptimo de umbrales tanto en las imágenes sintéticas como en las naturales. De acuerdo con Zheng (2016), el número óptimo de umbrales para cada imagen de la figura 6.3 es determinado de acuerdo a un grupo de expertos que manualmente delimitan las regiones.

Tabla 6.1. Umbrales seleccionados por cada algoritmo.

Image	nt óptimo (Zheng et al., 2016)	ATMO (Zheng et al., 2016)	AMT_ME (Zheng et al., 2016)	UTH-NSGA III	UTH-MOPSO	UTH-PESA-II
Mmi3	3	63-101-184	55-112-227	52-142-202	41-108-179	51-117-203
Mmi5	5	30-101-123-171-224	22-86-139-174-231	19-53-88-141-198	37-80-123-180-221	40-79-125-180-226
Rice	1,2	114-180	78-143	77-130	99-135	77-130
Tape	2,3	40-90-124	60-192	49-90-155	31-103-177	49-90-155
Cell	1,2	98	76-137	93-135	93-120	92-135

Puesto que todos los métodos son capaces de identificar el número correcto de umbrales requeridos, el siguiente paso es analizar la calidad de cada conjunto de valores umbral. Con este fin, las métricas de calidad enumeradas en la subsección 2.6 se evalúan teniendo en cuenta los umbrales de la Tabla 6.1. Los resultados de la comparación de calidad entre estos cinco métodos se pueden encontrar en la Tabla 6.2, donde el mejor valor de cada métrica para cada imagen está marcado como negrita. Es fácil detectar que los enfoques basados en UTH funcionan mejor que ATMO y AMT-ME, siendo NSGA III la metodología UTH de mejor desempeño.

Tabla 6.2. Comparación con enfoques anteriores.

Image	ATMO			AMT_ME			UTH-NSGA III			UTH-MOPSO			UTH-PESA-II		
	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
Mmi3	15.8260	0.6523	0.7827	17.4839	0.6540	0.7822	18.1767	0.6620	0.7807	14.2454	0.6139	0.7781	16.6177	0.6497	0.7830
Mmi5	18.4046	0.6335	0.7438	20.8692	0.8512	0.7745	21.2652	0.8675	0.7750	20.0342	0.7542	0.8022	20.5031	0.7585	0.8021
Rice	10.8141	0.2243	0.6870	16.2485	0.5650	0.7030	16.3988	0.5748	0.7156	12.4546	0.2953	0.6404	16.3988	0.5748	0.7156
Tape	18.8595	0.4582	0.8028	16.3598	0.3598	0.7886	19.1572	0.4585	0.8191	17.0387	0.4172	0.7606	19.1572	0.4585	0.8191
Cell	18.9766	0.8084	0.7051	17.0038	0.8148	0.7600	19.6835	0.8384	0.7552	19.6273	0.8383	0.7621	19.6138	0.8390	0.7570

6.2.5.4 Análisis de convergencia

En esta subsección se evalúa la convergencia de UTH. Para este propósito, UTH se implementa en tres MOEAs como UTH-NSGA-II, UTH-MOPSO, y UTH-PESA-II. La convergencia de UTH debe ser evaluada para determinar si las implementaciones propuestas de UTH pueden encontrar soluciones óptimas para el problema de la umbralización. La Tabla 6.3 reporta el valor de aptitud (entropía cruzada) obtenido por la partícula seleccionada mediante el proceso de decisión descrito en la subsección 6.2.4. En la Tabla 6.3, el algoritmo UTH-MOPSO reporta mejores valores de aptitud. Sin embargo, como se indica en el Cuadro 6.4, el algoritmo UTH-MOPSO genera soluciones con nt grandes, proporcionando soluciones con ocho umbrales en promedio, mientras que las combinaciones UTH-NSGA-III y UTH-PESA-II proporcionan menos umbrales. Así, el propósito UTH es encontrar un balance entre número de umbrales nt y calidad de la segmentación.

Tabla 6.3. Valor de primero objetivo y desviación estándar.

Image	UTH-NSGA III			UTH-MOPSO			UTH-PESA-II		
Boat	0.7877	(3.04E-01)	0.63747	(4.23E-01)	1.5348	(2.98E-01)			
Cameraman	0.8602	(4.80E-01)	0.71705	(6.49E-01)	2.5956	(7.22E-01)			
Jet	0.4405	(1.78E-01)	0.28974	(1.74E-01)	0.9985	(2.46E-01)			
Lena	0.6962	(4.24E-01)	0.58206	(3.26E-01)	1.4993	(3.54E-01)			
Living room	1.0238	(5.11E-01)	0.67034	(3.69E-01)	1.6524	(3.81E-01)			
Peppers	1.1767	(4.50E-01)	0.71333	(5.78E-01)	1.9271	(2.73E-01)			
Pirate	0.6953	(3.83E-01)	0.54622	(3.32E-01)	1.4040	(4.17E-01)			
Sailboat on a lake	0.8293	(3.59E-01)	0.49106	(3.50E-01)	2.0582	(8.14E-01)			
Splash	1.3522	(7.07E-01)	0.79196	(5.75E-01)	2.0841	(4.35E-01)			
Woman Blondie	0.6661	(3.40E-01)	0.37808	(1.81E-01)	1.4017	(3.01E-01)			

El número promedio de umbrales nt se presenta junto a los valores normalizados utilizados como segunda función objetivo, junto con la desviación estándar de cada objetivo entre paréntesis. Los resultados indican que el UTH-PESA-II es capaz de encontrar las partículas con menos valores umbral. Sin embargo, de acuerdo con la Tabla 6.3, las partículas encontradas por UTH-PESA-II son las peores partículas con respecto a la entropía cruzada. Por otra parte, todas las partículas de UTH-MOPSO que obtienen los mejores resultados en la Tabla 6.3 son las soluciones que incluyen el mayor número de umbrales en promedio en la Tabla 6.4. Este hecho apoya la afirmación de que ambos objetivos son conflictivos. Por lo tanto, el enfoque UTH es adecuado para las MOEA.

Tabla 6.4. Valor del segundo objetivo y su desviación estándar.

Image	UTH-NSGA III			UTH-MOPSO			UTH-PESA-II		
	nt	Normalizado	σ	nt	Normalizado	σ	nt	Normalizado	σ
Boat	4.24	0.13281	(7.52E-02)	6.16	0.19271	(7.82E-02)	2.34	0.072917	(3.08E-02)
Cameraman	3.55	0.11111	(6.13E-02)	5.83	0.18229	(6.58E-02)	1.41	0.044271	(2.16E-02)
Jet	3.80	0.11892	(5.00E-02)	6.47	0.20226	(6.27E-02)	2.33	0.072917	(3.95E-02)
Lena	4.38	0.13715	(9.03E-02)	5.57	0.17274	(5.15E-02)	2.41	0.075521	(3.11E-02)
Living room	4.30	0.13455	(8.04E-02)	6.11	0.19097	(5.43E-02)	2.58	0.080729	(2.73E-02)
Peppers	3.44	0.10764	(5.04E-02)	6.44	0.20139	(7.03E-02)	2.22	0.069444	(1.85E-02)
Pirate	3.77	0.11806	(5.74E-02)	5.77	0.18056	(5.60E-02)	2.41	0.075521	(3.29E-02)
Sailboat on a lake	3.94	0.12326	(6.11E-02)	6.91	0.21615	(6.44E-02)	2.08	0.065104	(4.69E-02)
Splash	3.55	0.11111	(6.09E-02)	5.86	0.18316	(6.21E-02)	2.44	0.076389	(2.64E-02)
Woman Blondie	4.27	0.13368	(6.75E-02)	7.16	0.22396	(5.81E-02)	2.27	0.071181	(1.42E-02)

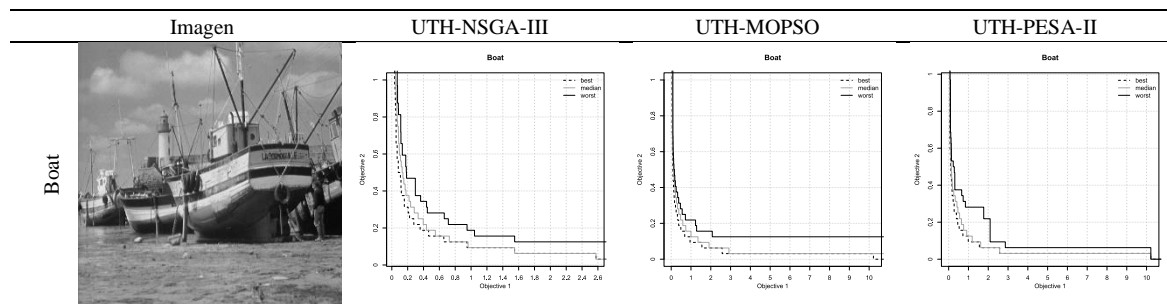
El indicador de hipervolumen describe el desempeño de cobertura de una MOEA. Dicha métrica se computa como el hipervolumen generado desde un único punto hacia el conjunto no dominado de partículas generadas por un algoritmo multiobjetivo. En el escenario de minimización, un punto se da de tal manera que sus coordenadas son mayores que los valores máximos de cada función objetivo. En este trabajo se selecciona el punto (3, 1) como se indica en la subsección 6.2.5.1. Los valores reportados de hipervolumen se muestran en la Tabla 6.5. En las comparaciones de hipervolumenes, el UTH-NSGA-III supera a los otros enfoques por sus mayores valores frente a los otros métodos, indicando que el UTH-NSGA-III puede encontrar soluciones no dominadas más cercanas al mínimo con una desviación estándar más baja.


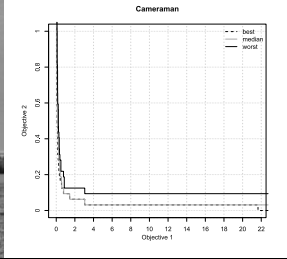
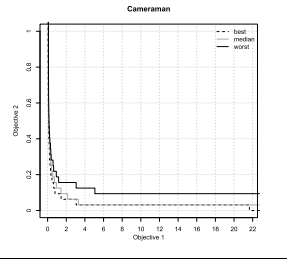
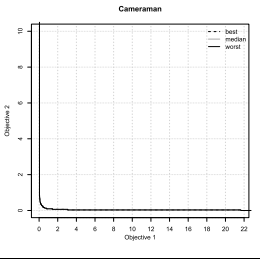

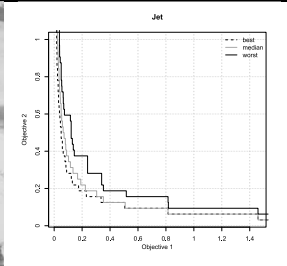
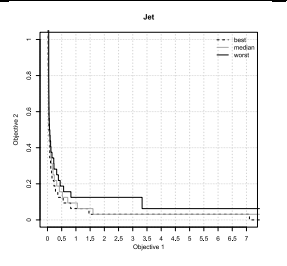
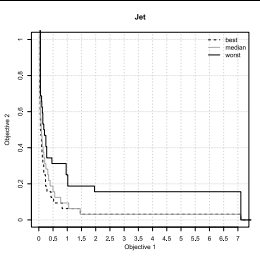

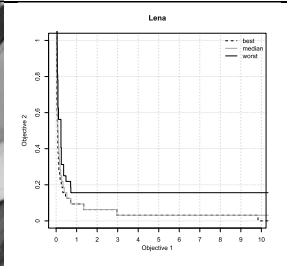
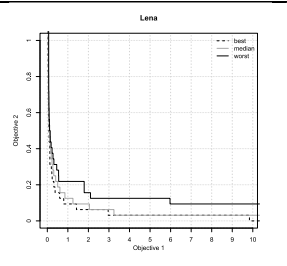
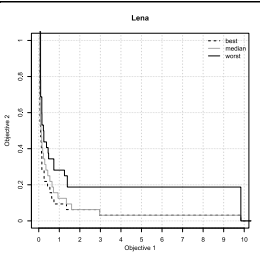

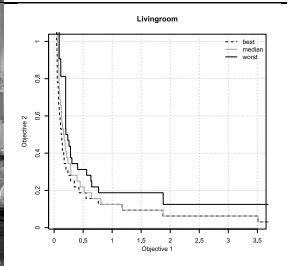
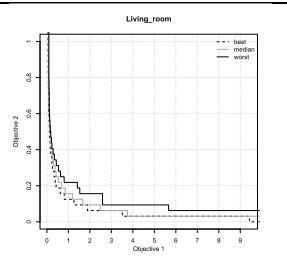
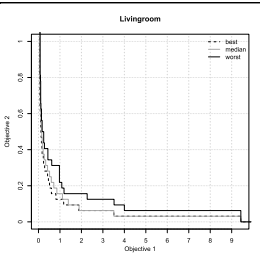

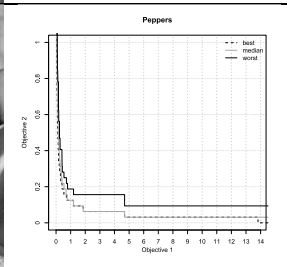
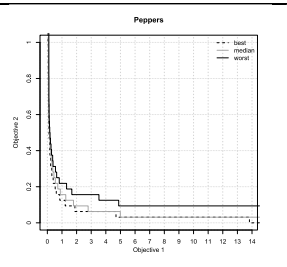
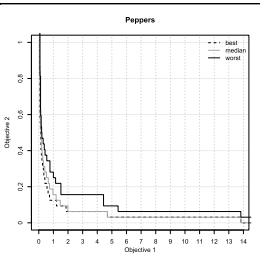

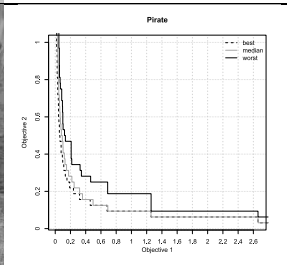
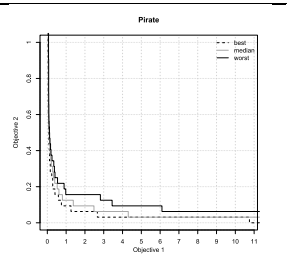
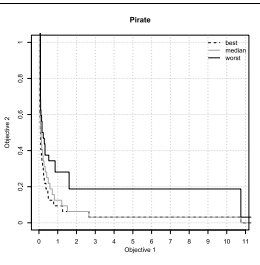
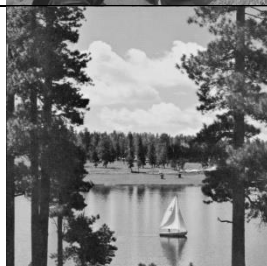
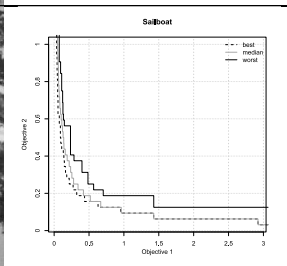
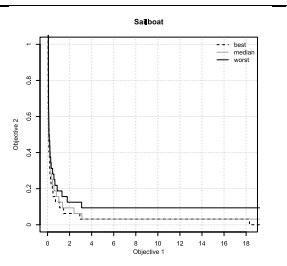
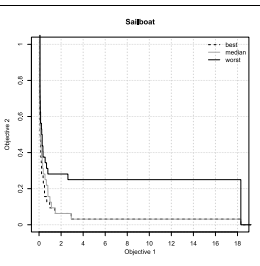
Como se ha indicado previamente, el desempeño de los MOEAs es a menudo evaluado con diferentes métricas, donde el hipervolumen es uno de ellos. Sin embargo, la mayoría de esas métricas requieren que el Frente de Pareto Verdadero sea conocido para medir la distancia de las soluciones dadas hacia las soluciones verdaderas. En la UTH el verdadero frente de Pareto es desconocido. Para abordar este tema, se selecciona el hipervolumen para analizar la convergencia de las tres implementaciones de UTH. Aún más, esta convergencia se evalúa visualmente usando los gráficos de superficie de logro de la Figura 6.4. Dichas superficies trazan tres conjuntos de soluciones no dominadas; la mejor, la mediana y la peor de 36 evaluaciones de cada método. En caso de minimización se espera que las tres curvas estén lo más cerca posible del origen.

Tabla 6.5. Hipervolumen y desviación estándar.

Image		UTH-NSGA III	UTH-MOPSO	UTH-PESA-II
Boat	0.85053	(1.03E-02)	0.84172 (7.86E-03)	0.84060 (2.18E-02)
Cameraman	0.84848	(1.11E-02)	0.83575 (1.19E-02)	0.83525 (2.45E-02)
Jet	0.90562	(5.53E-03)	0.89273 (1.26E-02)	0.87962 (2.85E-02)
Lena	0.86205	(1.80E-02)	0.84478 (1.41E-02)	0.83692 (2.60E-02)
Living room	0.83240	(1.04E-02)	0.81984 (9.67E-03)	0.82441 (1.56E-02)
Peppers	0.83383	(1.01E-02)	0.81467 (1.26E-02)	0.81997 (1.56E-02)
Pirate	0.87566	(6.51E-03)	0.84706 (1.21E-02)	0.84375 (3.18E-02)
Sailboat on a lake	0.85376	(8.70E-03)	0.83580 (1.04E-02)	0.82174 (3.84E-02)
Splash	0.83277	(9.33E-03)	0.81714 (1.21E-02)	0.81408 (2.15E-02)
Woman Blondie	0.86153	(7.77E-03)	0.85157 (1.14E-02)	0.85971 (1.33E-02)

Como se muestra en la Figura 6.4, el UTH-NSGA-III presenta una mejor convergencia en las superficies de logro. Este resultado y el hipervolumen apoyan la sugerencia de NSGA-III para UTH. En muchos de los gráficos, el eje del primer objetivo se amplía para que en el NSGA-III pueda apreciar correctamente el comportamiento de convergencia. Por ejemplo, imágenes como boat, jet, living room, pirata, sailboat on a lake, y splash toman valores de hasta tres, mientras que los otros enfoques muestran gráficos ampliados a valores de hasta dieciocho (UTH-PESA-II, sailboat). La implementación de UTH-MOPSO genera superficies de logro consistentes, mientras que la versión UTH-PESA-II presenta un rendimiento inferior en la mayoría de las imágenes teniendo como excepción a cameraman.



Cameraman				
Jet				
Lena				
Living room				
Peppers				
Pirate				
Sailboat on a lake				

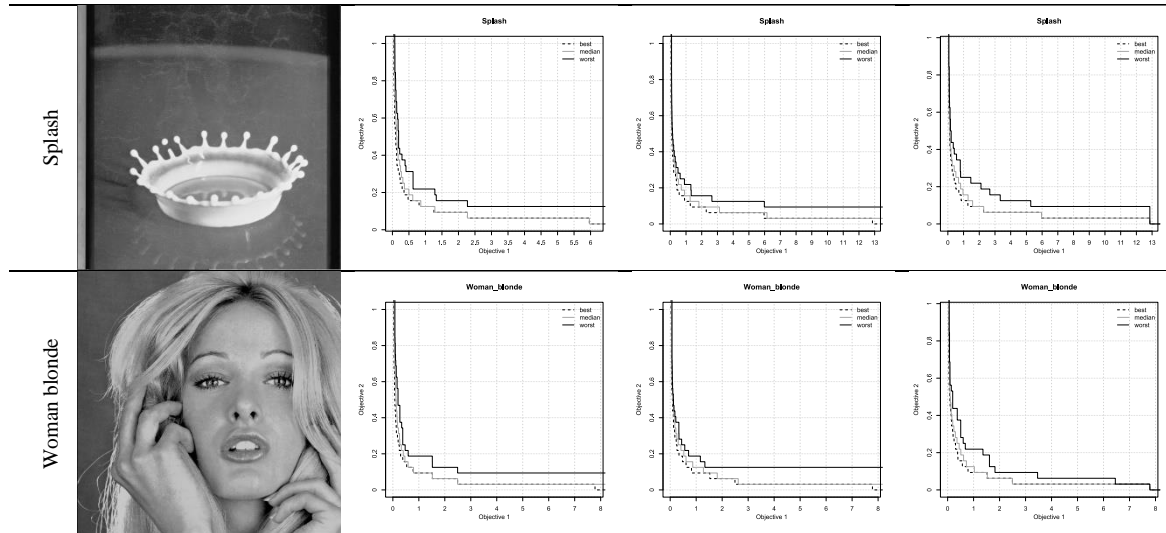


Figura 6.4. Superficies de logro.

6.2.5.5 Análisis de calidad

El análisis anterior indica que NSGA-III puede proporcionar mejores conjuntos de soluciones para el problema de UTH. Sin embargo, debe abordarse la calidad de las soluciones propuestas para determinar si pueden proporcionar un rendimiento competente en materia de establecimiento de umbrales. La comparación entre técnicas se realiza con medidas de similitud como el PSNR, SSIM y FSIM y los resultados se muestran en las Tablas 6.6, 6.8 y 6.9 respectivamente. En la Tabla 6.6 se presentan los umbrales generados por cada versión de la implementación de UTH. Como cada MOEA genera como salida un conjunto de umbrales, se evalúa el proceso de toma de decisiones para seleccionar qué partícula se utilizará para la segmentación. Los umbrales reportados en la Tabla 6.9 corresponden a los umbrales seleccionados por la toma de decisiones del proceso que apareció con mayor frecuencia en la experimentación.

Tabla 6.6. Los mejores umbrales encontrados por cada MOEA.

Image	UTH-NSGA-III	UTH-MOPSO	UTH-PESA-II
Boat	57 114 163	33 54 61 86 119 133 149 179	66 135
Cameraman	30 83 144	19 33 65 91 107 131 157	52 136
Jet	79 130 181	65 98 117 146 184 189 205 206	96 167
Lenna	73 122 168	45 69 88 119 143 174 190	83 139
Livingroom	38 83 124 161	40 50 78 90 114 148 181	77 132
Peppers	58 111 162	34 65 98 119 142 159 194	66 126
Pirate	59 89 122 158	17 62 91 97 125 147 181	64 123
Sailboat	57 89 144 196	36 52 80 122 160 176 203 222	75 145
Splash	45 87 153	26 59 94 116 135 162 209	31 83 158
Blonde	49 112 159	37 40 61 98 120 143 157 179	36 98 152

En cuanto a la calidad, las imágenes se pueden comparar directamente, independientemente de los criterios no paramétricos seleccionados como primera función objetivo. Así, las tablas del 6.7 a 6.9 comparan directamente los resultados de cada variante de la UTH propuesta. En términos de PSNR (Tabla 6.7), el UTH-NSGA-III supera a cualquier otra versión. Esta tendencia continúa en la Tabla 6.8, donde el Índice de Similitud Estructural

(SSIM), descrito en la sección 2.5.4. indica que la versión de UTH combinada con NSGA-III y Entropía Cruzada genera imágenes segmentadas donde las estructuras de la imagen se conservan mejor.

Tabla 6.7. Valores de la relación señal/ruido pico (PSNR) para cada una de las desviaciones estándar de la técnica.

Image	UTH-NSGA III		UTH-MOPSO		UTH-PESA-II	
Boat	19.9809	(2.72E+00)	17.946	(5.38E+00)	16.3067	(1.52E+00)
Cameraman	19.3416	(2.86E+00)	18.5851	(5.74E+00)	11.7883	(2.55E+00)
Jet	20.0213	(2.78E+00)	16.7872	(8.27E+00)	13.685	(1.93E+00)
Lena	18.3592	(2.52E+00)	16.8337	(4.82E+00)	15.4733	(1.06E+00)
Living room	20.2472	(3.38E+00)	19.4523	(4.74E+00)	16.8123	(1.23E+00)
Peppers	18.6203	(2.61E+00)	17.619	(5.67E+00)	15.1582	(1.40E+00)
Pirate	18.3088	(1.73E+00)	18.7092	(3.01E+00)	15.9809	(1.51E+00)
Sailboat on a lake	17.7284	(2.60E+00)	17.7353	(5.81E+00)	13.2962	(2.30E+00)
Splash	18.4379	(3.09E+00)	18.1868	(4.93E+00)	15.6669	(9.11E-01)
Woman Blondie	20.2114	(3.50E+00)	19.0802	(6.28E+00)	14.7886	(1.24E+00)

Tabla 6.8. Índice de Similitud Estructural (SSIM) y desviación estándar.

Image	UTH-NSGA-III		UTH-MOPSO		UTH-PESA-II	
Boat	0.65281	(7.99E-02)	0.61388	(2.17E-01)	0.55691	(4.91E-02)
Cameraman	0.6426	(5.13E-02)	0.60906	(2.07E-01)	0.48943	(9.28E-02)
Jet	0.79689	(2.52E-02)	0.63578	(2.87E-01)	0.71888	(4.71E-02)
Lena	0.6403	(7.79E-02)	0.62649	(1.20E-01)	0.57963	(4.31E-02)
Living room	0.6935	(9.91E-02)	0.68738	(1.52E-01)	0.58435	(5.56E-02)
Peppers	0.63291	(5.21E-02)	0.62213	(1.40E-01)	0.58151	(2.94E-02)
Pirate	0.58708	(7.65E-02)	0.63374	(9.68E-02)	0.5109	(7.05E-02)
Sailboat on a lake	0.61822	(7.91E-02)	0.65773	(1.79E-01)	0.49597	(9.93E-02)
Splash	0.65373	(6.38E-02)	0.66588	(1.02E-01)	0.60182	(4.54E-02)
Woman Blonde	0.71396	(7.93E-02)	0.66327	(1.92E-01)	0.57584	(4.00E-02)










En la Tabla 6.9 se reporta el Índice de Similitud de las Características (FSIM) descrito en la sección 2.5.5 para evaluar la calidad de las imágenes segmentadas. Como se ha indicado anteriormente, los resultados se generan tras 36 evaluaciones de cada algoritmo en cada imagen. Aunque UTH-MOPSO utiliza más valores umbral en cada segmentación, los resultados de UTH-NSGA-III indican una mejor calidad con menos umbrales. La métrica FSIM indica que la calidad de las imágenes segmentadas usando NSGA-III y la formulación UTH es superior a los otros enfoques evaluados. Dado que las tres métricas de calidad (PSNR, SSIM y FSIM) indican que el UTH-NSGA-III genera mejores resultados, así es posible sugerir el uso del NSGA-III para el UTH.

Tabla 6.9. Índice de similitud de las características (FSIM) y desviación estándar.

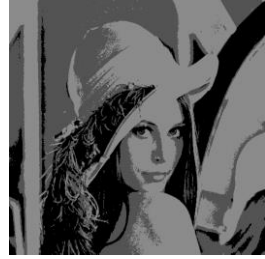
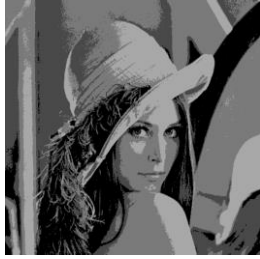
Image	UTH-NSGA-III		UTH-MOPSO		UTH-PESA-II	
Boat	0.87064	(4.23E-02)	0.84831	(8.16E-02)	0.76427	(4.24E-02)
Cameraman	0.85168	(4.65E-02)	0.85417	(7.77E-02)	0.72428	(5.57E-02)
Jet	0.87818	(3.49E-02)	0.81628	(1.31E-01)	0.77964	(2.69E-02)
Lena	0.83237	(5.20E-02)	0.82352	(5.77E-02)	0.7639	(2.51E-02)

Living room	0.85381	(6.68E-02)	0.86586	(6.76E-02)	0.77691	(3.52E-02)
Peppers	0.81626	(4.78E-02)	0.8303	(7.05E-02)	0.75081	(2.07E-02)
Pirate	0.85736	(4.81E-02)	0.85935	(4.98E-02)	0.78325	(3.70E-02)
Sailboat on a lake	0.87761	(2.60E-02)	0.87905	(6.59E-02)	0.79313	(5.32E-02)
Splash	0.80022	(5.15E-02)	0.83294	(5.13E-02)	0.76192	(2.06E-02)
Woman Blondie	0.85786	(6.77E-02)	0.84723	(9.08E-02)	0.72335	(4.01E-02)

Visualmente, la Figura 6.5 muestra las imágenes segmentadas considerando los valores de la Tabla 6.4. La mayoría de las imágenes pueden describirse como competitivas. Sin embargo, un análisis cualitativo es difícil ya que algunas imágenes se generan con más intensidad que otras. Por ejemplo, el velero en un lago (*sailboat*) utiliza ocho umbrales para UTH-MOPSO, cuatro en UTH-NSGA-III y dos con UTH-PESA-II. La diferencia es bastante notable entre las tres imágenes segmentadas. De acuerdo con el análisis de convergencia y calidad, la segmentación realizada por NSGA-III en UTH proporciona un mejor equilibrio entre la calidad de la imagen segmentada y el número de umbrales utilizados para dicha segmentación.

	UTH-NSGA-III	UTH-MOPSO	UTH-PESA-II
Boat			
Cameraman			
Jet			

Lena



Livingroom



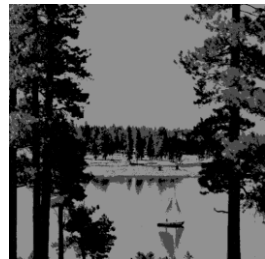
Peppers



Pirate



Sailboat on a lake



Splash

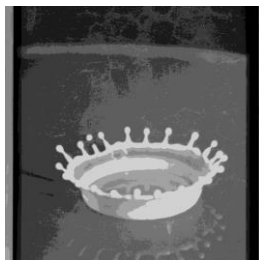




Figura 6.5. Comparación cualitativa del enfoque UTH.

6.2.6 Discusión

En este trabajo se adaptan tres MOEAs para determinar el número óptimo de umbrales para la segmentación de imágenes usando la técnica de entropía cruzada. De acuerdo con las Tablas 6.4 y 6.5, la formulación de la UTH es adecuada para los MOEA, ya que ambas funciones objetivo son conflictivas. Con las 36 ejecuciones individuales, el método UTH-NSGA-III propuesto mostró un mejor desempeño y consistencia con respecto a PSNR, SSIM y FSIM. Este enfoque demuestra ser una opción interesante para determinar el número requerido de umbrales sin comprometer la similitud de la imagen original.

Sin embargo, los resultados de la Tabla 6.4 muestran que el número seleccionado de umbrales depende en gran medida de la MOEA evaluada. Este factor puede deberse a los mecanismos de búsqueda que intervienen en el proceso de búsqueda. Dado que el algoritmo MOPSO no tiene un operador de cruce, la implementación consideró umbrales ficticios; este tipo de implementación podría sesgar las soluciones hacia partículas con un gran número de umbrales. Los resultados muestran efectivamente que UTH-MOPSO genera soluciones con más umbrales que UTH-NSGA-III y UTH-PESA-II. Los puntos de referencia de UTH-NSGA-III pueden beneficiar al algoritmo para encontrar mejores soluciones no dominadas. Como resultado, los Frentes de Pareto generados por NSGA-III (Figura 6.4) funcionan mejor que los enfoques comparados. En cuanto a la calidad, las imágenes segmentadas son visualmente muy similares; sin embargo, las métricas cuantitativas indican que el UTH-NSGA-III supera cualquier otro método. Sorprendentemente, los resultados generados por UTH-NSGA-III tienen menos umbrales que los de UTH-MOPSO; puesto que UTH-MOPSO produce más umbrales, la calidad debería ser más alta. Este hecho sugiere que UTH-MOPSO no es capaz de encontrar buenos umbrales para el problema de la UTH. El algoritmo PESA-II se centra en menos umbrales, pero también con una calidad deficiente, según se informa tanto en el análisis de convergencia como en el de calidad. En este contexto, se fomenta el uso de NSGA-III para UTH.

6.3 Mejora de algoritmos multiobjetivo usando caos.

En esta sección se proponen dos versiones multiobjetivo respecto del recientemente publicado algoritmo de búsqueda del cuervo (CSA) (Askarzadeh, 2016). Los operadores de CSA son mejorados tanto para las etapas de exploración como de explotación. En uno de los enfoques, la exploración se mejora mediante la incorporación de una longitud de vuelo caótico basado en el mapa caótico Tent (Caponetto, Fortuna, Fazzino, & Xibilia, 2003). Dado que CSA fue propuesto como una técnica de optimización de un único objetivo, es necesario añadir nuevos mecanismos al CSA para considerar múltiples objetivos. La incorporación de un archivo para almacenar soluciones no dominadas junto con una rejilla adaptativa destinada a mantener la diversidad de las soluciones da lugar a la propuesta. Dado que CSA utiliza mecanismos simples y tiene desempeño aceptable, la versión multiobjetivo tiene como finalidad proporcionar una técnica competitiva a la vez que preserva su simplicidad de implementación original. La propuesta se discute en dos subsecciones. La primera realiza la incorporación de mecanismos multiobjetivo al CSA generando una versión llamada *Multi-Objective CSA* (MOCSA). La segunda subsección toma al algoritmo MOCSA y lo mejora mediante la incorporación de series caóticas, lo que lleva a la propuesta del algoritmo *Multi-Objective Chaotic Crow Search Algorithm* (MOCCSA) (Hinojosa et al., 2017).

6.3.1 CSA multiobjetivo

La descripción del algoritmo CSA puede encontrarse en la primera sección del capítulo 4. Así, la expresión para generar nuevas soluciones se mejora como se describe en la Ec. 6.7:

$$\mathbf{c}_i^{k+1} = \begin{cases} \mathbf{c}_i^k + r \times fl \times (\mathbf{h}_j^k - \mathbf{c}_i^k) & r_j \geq AAP^k \\ \mathbf{c}_i^k + r \times fl \times (\mathbf{d}_j^k - \mathbf{h}_i^k) & \text{en otro caso} \end{cases} \quad (6.7)$$

$$fl = cte$$

$$AAP^k = \frac{k}{\text{Max Iteration}}$$

donde r y r_j son números aleatorios generados mediante una distribución uniforme entre 0 y 1, fl es la longitud de vuelo que debe ser establecida como una constante. El término \mathbf{h}_j^k indica un escondite de comida el cual es seleccionado mediante un operador de ruleta, \mathbf{d}_j^k es una posición de engaño generada aleatoriamente dentro de los límites del espacio de búsqueda mientras que AAP^k es la probabilidad de conocimiento adaptativa en la iteración k . El algoritmo 6.1 muestra el pseudocódigo del algoritmo MOCSA.

-
- 1.- Inicializa población de cuervos \mathbf{C}^0
 - 2.- Determina el valor de aptitud de cada cuervo $f(\mathbf{C}^0)$
 - 3.- Determina dominancia
 - 4.- Inicializa el archivo con soluciones no dominadas
 - 5.- Repite
 - 6.- Para cada $\mathbf{c}^k \in \mathbf{C}^k$
 - 7.- Si $(rand < AAP^k)$
 - 8.- selecciona \mathbf{h}^k mediante operador de ruleta
 - 9.- En otro caso
 - 10.- Posición de engaño \mathbf{d}_j^k generada aleatoriamente
 - 11.- fin
 - 12.- El cuervo \mathbf{c}^k se mueve de acuerdo con (Ec.6.7)
 usando $fl = cte$
 - 13.- fin
 - 14.- Evaluar de aptitud de cada cuervo $f(\mathbf{C}^k)$
 - 15.- Encuentra soluciones no dominadas
 - 16.- Actualiza archivo
 - 17.- Hasta $k > MaxIt$
 - 18.- Regresa archivo
-

Algoritmo 6.1. Pseudocódigo del algoritmo *Multi-Objective Crow Search* (MOCSA).

Debido a la naturaleza poblacional del algoritmo CSA, se puede aplicar la estrategia usada por MOPSO en este trabajo (Coello et al., 2004). CSA es convertido a MOCSA mediante la implementación de una memoria denominada archivo y una estrategia de selección. El archivo almacena las soluciones de Pareto no dominadas obtenidas en cada iteración, en este caso, los escondites de alimentos (Algoritmo 6.1 paso 4 y 16). El archivo se llena en primer lugar con las soluciones no dominadas generadas por el proceso de inicialización. Por lo tanto, el número de elementos en el archivo, después de una inicialización aleatoria, no puede determinarse *a priori*. Esta unidad de almacenamiento es finita y requiere un controlador que impida que el archivo esté lleno. Para evitar la redundancia o la presencia de soluciones muy similares, se adopta un mecanismo de rejilla adaptativa. La estrategia de selección utilizada funciona dando a las áreas menos pobladas mayores posibilidades de ser seleccionadas para mejorar la cobertura del frente de Pareto. Para ello, se seleccionan los peores y mejores valores de aptitud del conjunto de soluciones no dominadas existentes en ese momento para definir un hiperespacio que incluya todas las soluciones actuales. Posteriormente, el espacio se divide por una cuadrícula de tamaño adaptable donde cada celda contiene un número finito de soluciones. Un operador de ruleta se utiliza para determinar qué cuadrícula se considera que ofrece mayores posibilidades en las celdas menos pobladas de la cuadrícula. El mecanismo de rejilla adaptativa tiene por objeto proporcionar diversas soluciones. Utilizando esta técnica, el espacio de búsqueda se segmenta en varias regiones. Cuando una nueva solución cae fuera de la rejilla actual, las posiciones de la rejilla se ajustan para cubrir todas las soluciones. Además, cuando

una nueva solución aparece dentro de los límites de la cuadrícula, se dirige a la región de la cuadrícula menos densamente poblada. Para mantener la coherencia sobre las soluciones presentes en el archivo se siguen las reglas sugeridas por Coello (2004). La primera es que sólo se pueden incluir en el archivo las soluciones no dominadas de la población actual. Si un nuevo elemento del archivo domina las soluciones anteriores, las soluciones dominadas se descartan del archivo. Por último, si el archivo está lleno, se eliminan una o más soluciones de las regiones más pobladas de la rejilla.

6.3.2 CSA caótico y multiobjetivo

Para analizar los beneficios de la incorporación de series caóticas en el algoritmo de optimización, en esta investigación se presentan dos enfoques similares: MOCCSA y su contraparte libre de caos MOCSA. La subsección 6.3.1 presentó a MOCSA como la incorporación directa de mecanismos multiobjetivo como la rejilla adaptativa y el archivo en el CSA. En esta subsección, el algoritmo *Multi-objective Chaotic Crow Search Algorithm* (MOCCSA) se presenta como la mejora de MOCSA mediante la integración de una serie caótica en el proceso.

Los mapas caóticos son ampliamente utilizados en algoritmos meteheurísticos y pueden definirse como sistemas dinámicos discretos descritos como:

$$cp_{k+1}^i = f(cp_k^i), \quad i=1,2,3,\dots,n \quad (6.8)$$

donde n es la dimensión del mapa; $f(cp_k^i)$ es la función que genera el mapa caótico. En esta investigación se usará el mapa caótico de Tent, el cual se presenta en la Ec. 6.9:

$$p_{k+1} = G(p_k) \quad (6.9)$$

$$G(p) = \begin{cases} \frac{p}{0.7}, & p < 0.7 \\ \frac{1}{0.3}p(1-p) & \text{de otra forma} \end{cases}$$

La principal diferencia entre MOCCSA y MOCSA radica en el hecho de que para MOCCSA la longitud del vuelo se calcula a partir de la ecuación de mapa caótico de Tent, Ec. 6.9, mientras que para MOCSA la longitud del vuelo fl_i^k es una constante definida por el usuario. El algoritmo 2 muestra la versión mejorada de MOCCSA. Hay que tener en cuenta que la principal diferencia entre ambos algoritmos se encuentra en el duodécimo paso. La ecuación de actualización para MOCCSA se establece en la Ec. 6.10.

$$\begin{aligned}
\mathbf{c}_i^{k+1} &= \begin{cases} \mathbf{c}_i^k + r \times fl_i^{k+1} \times (\mathbf{h}_j^k - \mathbf{c}_i^k) & r_j \geq AAP^k \\ \mathbf{c}_i^k + r \times fl_i^{k+1} \times (\mathbf{d}_j^k - \mathbf{h}_i^k) & \text{en otro caso} \end{cases} \\
fl_i^{k+1} &= G(fl_i^k) \\
AAP &= \frac{k}{\text{Max Iteration}}
\end{aligned} \tag{6.10}$$

La incorporación de series caóticas se realiza en la Ec. 6.10 donde el valor actual fl_i^{k+1} se genera a partir del mapa caótico de Tent $G(\cdot)$ y su valor anterior fl_i^k . El resultado es un número real positivo que toma valores en el intervalo $[0,1]$. El mapa Tent se utiliza para generar una serie caótica en la que para cada iteración k se utiliza el valor correspondiente de la serie como fl_i^{k+1} . La figura 6.6 muestra gráficamente el comportamiento de la longitud del vuelo (fl). Con la variación de dicho parámetro, el algoritmo puede realizar una mejor exploración que conduce a mayores posibilidades de encontrar el conjunto óptimo de Pareto. Por el contrario, una longitud de vuelo fija podría llevar el proceso de optimización hacia soluciones subóptimas. En el esquema multiobjetivo, el frente de Pareto generado por el algoritmo puede estancarse muy lejos del verdadero Pareto cuando la formulación del problema incluye discontinuidades y multimodalidad. Para evitar estos problemas, muchos algoritmos incorporan caminatas al azar, vuelos Levy y perturbaciones caóticas. El coste computacional del algoritmo no se ve afectado directamente por la integración de esta característica, ya que la serie caótica puede calcularse antes de la inicialización para un número máximo de iteraciones.

En este capítulo sólo se aborda la incorporación de series caóticas en un único parámetro. El algoritmo CSA original tiene dos parámetros para ser ajustados por el usuario, la probabilidad de conocimiento (AP) y la longitud del vuelo (fl). Como se ha indicado previamente, MOCCSA utiliza una probabilidad de conciencia adaptativa (AAP) que controla la exploración y explotación. Este factor da mayores posibilidades de exploración al principio de la ejecución y disminuye linealmente hasta el final del tiempo de ejecución. Este comportamiento asegura la convergencia del algoritmo (Mirjalili, 2016a). Además, el parámetro seleccionado para ser tratado como un valor caótico es la longitud del vuelo (fl). Este valor contribuye a mejorar el rendimiento global del algoritmo sin comprometer su convergencia.

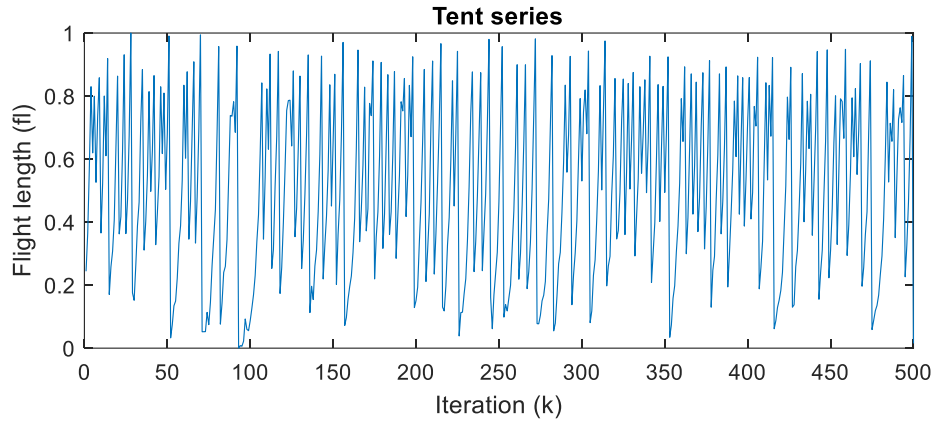


Figura 6.6. Comportamiento de la longitud de del vuelo (fl) duerante las iteraciones de MOCCSA.

En este caso se espera que la convergencia de MOCCSA sea la misma que la del algoritmo CSA regular, ya que los operadores de búsqueda son similares. En este contexto, cada solución seleccionada del archivo será mejorada por los mecanismos presentes en MOCCSA. La complejidad computacional de MOCCSA se evalúa como $O(MN^2)$ donde N es el número de elementos en la población y M es la cantidad de objetivos. La complejidad de MOCCSA es menor que la de otros algoritmos populares tal como tal SPEA y NSGA.

```

1.- Inicializa población de cuervos  $\mathbf{C}^0$ 
2.- Determina el valor de aptitud de cada cuervo  $f(\mathbf{C}^0)$ 
3.- Determina dominancia
4.- Inicializa el archivo con soluciones no dominadas
5.- Repite
6.-   Para cada  $\mathbf{c}^k \in \mathbf{C}^k$ 
7.-     Si ( $rand < AAP$ )
8.-       selecciona  $\mathbf{h}^k$  mediante operador de ruleta
9.-     En otro caso
10.-      Posición de engaño  $\mathbf{d}_j^k$  generada aleatoriamente
11.-     fin
12.-     El cuervo  $\mathbf{c}^k$  se mueve de acuerdo con (Ec. 6.10)
        usando  $fl^k = G(fl^{k-1})$ 
13.-   Fin
14.-   Evaluar de aptitud de cada cuervo  $f(\mathbf{C}^k)$ 
15.-   Encuentra soluciones no dominadas
16.-   Actualiza archivo
17.-   Hasta  $k > MaxIt$ 
18.-   Regresa archivo

```

Algoritmo 6.2. Pseudocódigo del algoritmo *Multi-Objective Chaotic Crow Search Algorithm* (MOCCSA).

6.3.3 Resultados experimentales

El algoritmo MOCCSA se compara con su versión libre de caos (MOCSA) y otros dos enfoques multiobjetivo, MOPSO y MODA (Coello et al., 2004; Mirjalili, 2016a). Los parámetros seleccionados para cada algoritmo se muestran en la Tabla 6.10. Los parámetros de

MOPSO y MODA se toman de sus respectivos artículos donde fueron publicados, mientras que MOCSA y MOCCSA toman parámetros de los dos algoritmos en los que se basan (CSA y MOPSO). La cuadrícula y los parámetros de archivo se toman de MOPSO y la longitud del vuelo, en el caso de MOCSA, se obtiene del algoritmo CSA.

Tabla 6.10. Parámetros seleccionados.

MOCCSA/MOCSA	MOPSO	MODA
Tamaño de la población: 100	Tamaño de la población: 100	Tamaño de la población: 100
Tamaño del archivo: 100	Tamaño del archivo: 100	Tamaño del archivo: 100
Rejillas por dimensión: 10	Rejillas por dimensión: 10	Max. Numero de hiper-esferas: 10
Taza de inflado: 0.1	Taza de inflado: 0.1	Peso de separación: 0.1
Presión de eliminación: 2	Presión de eliminación: 2	Peso de dirección: 0.1
(Sólo para MOCSA, longitud de vuelo: 2)	Presión del lider: 2	Peso de cohesión: 0.7
	Peso inercial: 0.5	Factor de comida: 1
	Taza de amortiguamiento del peso inercial: 0.99	Factor de enemigo: 1
	Coeficiente de aprendizaje personal: 3	Peso inercial: 0.9-0.2
	Coeficiente de aprendizaje social: 2	
	Taza de mutación: 0.1	

Por cada experimento, se utilizan 100 partículas y 3000 generaciones (k). Para la evaluación, se considera un conjunto de funciones de prueba estándar (CEC 2009) propuesto por Zhang (2009). Las funciones de prueba se enumeran y muestran en la Tabla 6.11. De todos los conjuntos de prueba ampliamente utilizados, este conjunto podría considerarse como el mejor de la literatura relacionada, ya que proporciona espacios de búsqueda con frentes óptimos de Pareto interesantes con propiedades como convexidad, no convexidad, discontinuidad y multimodalidad (Zhao et al., 2012). Una comparación cualitativa del enfoque propuesto se muestra en la Figura 6.8, donde el mejor frente de Pareto generado por cada algoritmo durante la experimentación se compara con el frente de Pareto real.

Tabla 6.11. Conjunto de pruebas CEC 2009.

Función	
UF1	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2, f_2 = 1 - \sqrt{x} + \frac{2}{ J_2 } \sum_{j \in J_2} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$ $J_1 = \{j \mid j \text{ es impar y } 2 \leq j \leq n\}, J_2 = \{j \mid j \text{ es par y } 2 \leq j \leq n\}$
UF2	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2, f_2 = 1 - \sqrt{x} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $J_1 = \{j \mid j \text{ es impar y } 2 \leq j \leq n\}, J_2 = \{j \mid j \text{ es par y } 2 \leq j \leq n\}$ $y_j = \begin{cases} x_j - \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right) & \text{if } j \in J_1 \\ x_j - \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) & \text{if } j \in J_2 \end{cases}$
UF3	$f_1 = x_1 + \frac{2}{ J_1 } \left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{f_1}}\right) + 2\right), f_2 = \sqrt{x_1} + \frac{2}{ J_2 } \left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{f_2}}\right) + 2\right)$ $J_1 = \{j \mid j \text{ es impar y } 2 \leq j \leq n\}, J_2 = \{j \mid j \text{ es par y } 2 \leq j \leq n\}$ $y_j = x_j - x_1^{0.5\left(1 - \frac{2(j-2)}{n-2}\right)}, j = 2, 3, \dots, n$
UF4	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j), f_2 = 1 - x^2 + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$

	$J_1 = \{j \mid j \text{ es impar y } 2 \leq j \leq n\}, J_2 = \{j \mid j \text{ es par y } 2 \leq j \leq n\}$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n, h(t) = \frac{ t }{1 + e^{2 t }}$
UF5	$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right) \sin(2N\pi x_1) + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j), f_2 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right) \sin(2N\pi x_1) + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $J_1 = \{j \mid j \text{ es impar y } 2 \leq j \leq n\}, J_2 = \{j \mid j \text{ es par y } 2 \leq j \leq n\}$ $\varepsilon > 0, y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n, h(t) = 2t^2 - \cos(4\pi t) + 1$
UF6	$f_1 = x_1 + \max\left\{0, 2\left(\frac{1}{2N} + \varepsilon\right) \sin(2N\pi x_1)\right\} + \frac{2}{ J_1 } \left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1\right)$ $f_2 = x_1 + \max\left\{0, 2\left(\frac{1}{2N} + \varepsilon\right) \sin(2N\pi x_1)\right\} + \frac{2}{ J_2 } \left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1\right)$ $J_1 = \{j \mid j \text{ es impar y } 2 \leq j \leq n\}, J_2 = \{j \mid j \text{ es par y } 2 \leq j \leq n\}$ $\varepsilon > 0, y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n$
UF7	$f_1 = \sqrt[3]{x_1} + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2, f_2 = 1 - \sqrt[3]{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $J_1 = \{j \mid j \text{ es impar y } 2 \leq j \leq n\}, J_2 = \{j \mid j \text{ es par y } 2 \leq j \leq n\}$ $\varepsilon > 0, y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n$
UF8	$f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $f_2 = \cos(0.5x_1\pi) \sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $f_3 = \sin(0.5x_2\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $J_1 = \{j \mid 3 \leq j \leq n, yj - 1 \text{ es múltiplo de } 3\}, J_2 = \{j \mid 3 \leq j \leq n, yj - 2 \text{ es múltiplo de } 3\}$ $J_3 = \{j \mid 3 \leq j \leq n, yj \text{ es múltiplo de } 3\}$
UF9	$f_1 = 0.5 \left[\max\left\{0, (1 + \varepsilon) \left(1 - 4(2x_1 - 1)^2\right)\right\} + 2x_1 \right] x_2 + \frac{2}{ J_1 } \sum_{j \in J_1} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $f_2 = 0.5 \left[\max\left\{0, (1 + \varepsilon) \left(1 - 4(2x_1 - 1)^2\right)\right\} + 2x_1 \right] x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $f_3 = 1 - x_2 + \frac{2}{ J_3 } \sum_{j \in J_3} \left(x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $J_1 = \{j \mid 3 \leq j \leq n, yj - 1 \text{ es múltiplo de } 3\}$ $J_2 = \{j \mid 3 \leq j \leq n, yj - 2 \text{ es múltiplo de } 3\}$ $J_3 = \{j \mid 3 \leq j \leq n, yj \text{ es múltiplo de } 3\} \quad \varepsilon = 0.1$
UF10	$f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} (4y_j^2 - \cos(8\pi y_j) + 1)$ $f_2 = \cos(0.5x_1\pi) \sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} (4y_j^2 - \cos(8\pi y_j) + 1)$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} (4y_j^2 - \cos(8\pi y_j) + 1)$ $J_1 = \{j \mid 3 \leq j \leq n, yj - 1 \text{ es múltiplo de } 3\}$

$$J_2 = \{j \mid 3 \leq j \leq n, \text{ y } j-2 \text{ es múltiplo de } 3\}$$

$$J_3 = \{j \mid 3 \leq j \leq n, \text{ y } j \text{ es múltiplo de } 3\}$$

Para analizar cuantitativamente el desempeño del enfoque propuesto, se seleccionan varias métricas de desempeño a saber: a) Distancia Generacional Invertida (IGD) (Sierra & Coello Coello, 2005) que mide la convergencia; b) Espaciado (SP) (Coello et al., 2004); c) la Dispersión Máxima (MS) (Eckart Zitzler & Thiele, 1999) que cuantifica la cobertura y d) la métrica Hipervolumen (Eckart Zitzler & Thiele, 1999) para evaluar el área/volumen del espacio de búsqueda dominado.

La métrica IGD se define como:

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (6.11)$$

donde n representa el número de soluciones no dominadas y d_i es la distancia euclidiana entre la i -ésima solución del frente de Pareto real y la solución no dominada más cercana obtenida por el algoritmo.

La métrica de espaciado SP se formula como:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (6.12)$$

$$d_i = \min \left(\left| f_1^i(\bar{x}) - f_1^j(\bar{x}) \right| + \left| f_2^i(\bar{x}) - f_2^j(\bar{x}) \right| \right) \text{ para } i, j = 1, 2, 3, \dots, n$$

donde \bar{d} es el promedio de todas las distancias d_i y n es el número de soluciones obtenidas. Así, la dispersión máxima es definida por la Ec. 6.13 .

$$MS = \sqrt{\sum_{i=1}^a \max(d(a_i, b_i))} \quad (6.13)$$

donde $d(.)$ es una función de distancia, a_i es el valor máximo en el objetivo i -ésimo, b_i es el mínimo valor en el i -ésimo objetivo y es el número de objetivos presentes en el problema.

Otra métrica es el hipervolumen diseñado por Zitzler (Eckart Zitzler, 1999). Cuantifica el comportamiento de convergencia de los algoritmos multiobjetivo. Como se ha indicado previamente, el principio subyacente es el cálculo del área, volumen o hipervolumen del espacio objetivo dominado por las soluciones de Pareto obtenidas. Para que esto funcione, se fija un punto en el espacio de búsqueda, y el hipervolumen se calcula desde el punto al frente. Debido a la naturaleza de las funciones de referencia, el punto fijo se fija como (3,3) para las funciones

bi-objetivo mientras que (3,3,3) para las tres funciones objetivo para todos los algoritmos. Estos valores de referencia son establecidos empíricamente ya que ningún objetivo de las funciones de prueba ha reportado un valor igual o superior a tres. Los valores generados por esta métrica son de 0 a 1 donde 1 indica el máximo hipervolumen posible desde el punto dado hacia el origen del espacio de búsqueda. Un valor más alto indica un mejor rendimiento.

Por último, Fonseca y Flemming (1996) introdujeron el término superficie de logro, que es un límite que separa el espacio objetivo en dos regiones: aquellas soluciones objetivas que son alcanzadas (o dominadas por o iguales a) los resultados generados por los algoritmos examinados. Los gráficos de superficie de logro indican la mejor mediana y la peor superficie de logro. Se espera que los tres estén lo más cerca posible del verdadero Pareto. La Figura 4 muestra la comparación de las superficies de logro para las funciones de referencia bi-objetivo.

6.3.4 Discusión

En la Tabla 6.12 se presenta el análisis estadístico de los valores de IGD generados por el MOCCSA y otros dos algoritmos multiobjetivo, MOPSO y MODA. La siguiente Tabla 6.13 se centra en la métrica SP y la posterior Tabla 6.14 presenta los valores de MS. Finalmente, la Tabla 6.15 muestra el comportamiento de la métrica de hipervolumen. En la Tabla 6.16 se presentan los resultados de una prueba estadística de significancia. Para una comparación cualitativa, la figura 6.8 muestra el mejor frente de Pareto obtenido por cada algoritmo y el óptimo de Pareto. Al final, la Figura 6.9 muestra las superficies de logro de cada algoritmo. Debe tenerse en cuenta que todos los resultados se obtienen a partir de 30 ejecuciones independientes de cada algoritmo con cada función de prueba como se ha mencionado previamente.

El objetivo de la primera comparación es determinar si la presencia de un parámetro basado en el caos mejora el rendimiento del algoritmo. Para ello, se recopilan y contrastan los datos estadísticos obtenidos a partir de las 30 evaluaciones de cada algoritmo. La figura 6.7 compara los valores de IGD proporcionados por MOCCSA y MOCSA. Los gráficos de caja muestran distribuciones más estrechas para los valores de IGD de MOCCSA, lo que indica resultados más consistentes. Además, los valores medios de MOCCSA son evidentemente más bajos que los de MOCSA para la mayoría de las funciones de prueba, las cuales llevan por nombre UF1 a UF10. El análisis estadístico de IGD enfatiza la convergencia de MOCCSA.

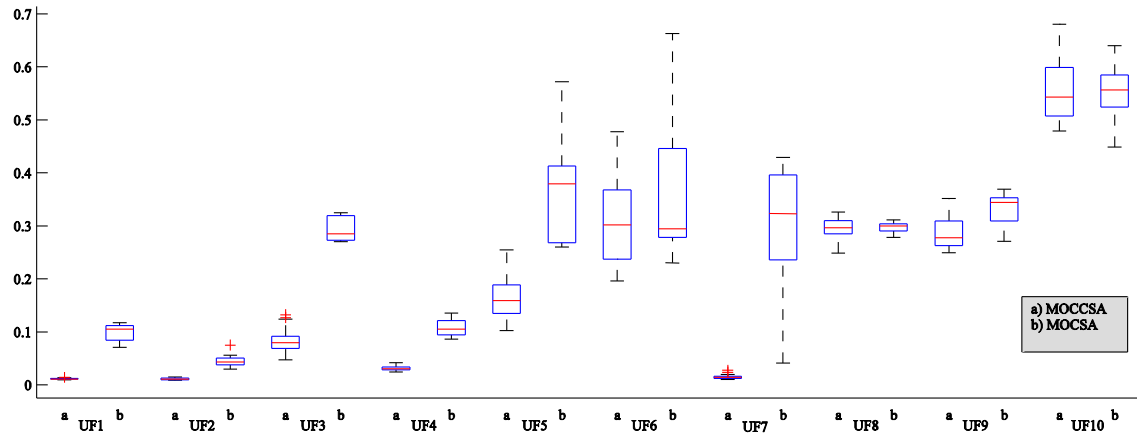


Figura 6.7. Comparación de gráficos de caja de los valores de IGD MOCCSA vs. MOCSA.

El IGD reportado por MOCCSA es consistentemente mejor que los valores generados por MOCSA, MOPSO y MODA. La Tabla 6.12 muestra una clara superioridad en el desempeño de MOCCSA con respecto a los valores IGD. Sólo UF6 muestra un valor medio de MOCSA ligeramente menor, que también puede apreciarse en la Figura 6.7. Estos resultados indican que el frente de Pareto generado por MOCCSA está más cerca del Pareto-óptimo que los enfoques comparados.

Tabla 6.12. Resultados estadísticos de la IGD para UF1 a UF10.

	UF1				UF2			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.0111	0.0987	0.0541	0.5106	0.0108	0.0459	0.0373	0.3004
Mediana	0.0110	0.1055	0.0520	0.5198	0.0104	0.0434	0.0374	0.2493
σ	0.0011	0.0171	0.0108	0.1274	0.0016	0.0128	0.0065	0.1153
Max	0.0140	0.1173	0.0753	0.7601	0.0144	0.0752	0.0502	0.5515
Min	0.0094	0.0709	0.0407	0.3432	0.0088	0.0295	0.0278	0.2040
	UF3				UF4			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.0834	0.2927	0.2400	0.6806	0.0308	0.1084	0.0540	0.1179
Mediana	0.0797	0.2848	0.2264	0.5065	0.0306	0.1054	0.0525	0.1177
σ	0.0224	0.0227	0.0708	0.5475	0.0037	0.0161	0.0061	0.0036
Max	0.1323	0.3243	0.4172	2.2156	0.0419	0.1356	0.0658	0.1246
Min	0.0471	0.2699	0.1312	0.3824	0.0242	0.0862	0.0437	0.1125
	UF5				UF6			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.1638	0.3692	0.2697	3.2674	0.3039	0.3533	0.4926	2.6141
Mediana	0.1587	0.3792	0.2461	3.1655	0.3017	0.2944	0.4530	2.6009
σ	0.0348	0.0978	0.1046	0.3142	0.0831	0.1304	0.1422	0.6879
Max	0.2543	0.5720	0.5819	3.9796	0.4780	0.6630	0.8437	3.5561
Min	0.1026	0.2599	0.1332	2.9809	0.1964	0.2301	0.2972	1.7096
	UF7				UF8			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.0148	0.2870	0.1316	0.4822	0.2960	0.2971	0.8691	0.7125
Mediana	0.0138	0.3227	0.1146	0.4977	0.2963	0.2998	0.8704	0.7118
σ	0.0039	0.1430	0.0537	0.1135	0.0220	0.0100	0.1132	0.1514
Max	0.0278	0.4290	0.2514	0.6220	0.3260	0.3111	1.1092	0.9058
Min	0.0103	0.0411	0.0595	0.3397	0.2488	0.2780	0.6436	0.4336
	UF9				UF10			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.2886	0.3301	0.9098	0.7029	0.5580	0.5691	5.3803	4.1110
Mediana	0.2778	0.3440	0.8905	0.6348	0.5429	0.5567	5.3169	4.3059
σ	0.0312	0.0348	0.1667	0.2503	0.0604	0.0599	0.6355	1.0752
Max	0.3519	0.3690	1.4467	1.3594	0.6808	0.6399	6.6449	5.2906
Min	0.2490	0.2705	0.6485	0.5111	0.4788	0.4484	4.2702	1.7200

Respecto de la Tabla 6.13, la métrica SP indica cuán cerca están las soluciones no dominadas entre sí. Se reporta un gran valor cuando muchas soluciones son conglomeradas. Por lo tanto, se prefiere un valor más pequeño. La Tabla 6.13 muestra resultados interesantes, aunque MOPSO obtuvo mejores resultados en UF4 y MODA en UF9, MOCCSA logra superar en la mayoría de los resultados. Por el contrario, MOCCSA reporta desviaciones estándar más pequeñas, lo que indica una alta consistencia de los resultados. Sorprendentemente, esta métrica es la única en la que MODA en UF9 muestra resultados competitivos.

La métrica de dispersión máxima (MS) también sugiere la superioridad de MOCCSA sobre MOCSA, MOPSO y MODA en la mayoría de las comparaciones. MS indica cuánto del Pareto-óptimo está cubierto por el algoritmo probado. En la Tabla 6.14, MOCCSA cubre porciones más grandes del Pareto-óptimo como lo indican los datos. Sin embargo, MOCSA muestra una σ más pequeña en dos de las diez funciones de prueba (UF1 y UF2), mientras que MOPSO califica la σ más baja en dos funciones. Los valores menores de σ de MOPSO y MOCSA indican que cubren consistentemente un área menor que MOCCSA. Asimismo, el indicador de hipervolumen señala una mayor cobertura de las soluciones dominantes de MOCCSA, como se indica en la Tabla 6.15. La mayoría de los valores indican un rendimiento superior de MOCCSA.

Tabla 6.13. Resultados estadísticos de SP para UF1 a UF10.

	UF1				UF2			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.0673	0.0600	0.1107	0.0964	0.0647	0.0438	0.0837	0.0748
Mediana	0.0671	0.0553	0.0506	0.0634	0.0630	0.0448	0.0482	0.0731
σ	0.0096	0.0225	0.1567	0.0855	0.0094	0.0088	0.1176	0.0878
Max	0.0846	0.1036	0.7628	0.2652	0.0861	0.0553	0.6417	0.2969
Min	0.0483	0.0348	0.0092	0.0216	0.0517	0.0280	0.0268	0.0308
	UF3				UF4			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.0321	0.0114	0.3294	0.0901	0.0627	0.0560	0.0469	0.0671
Mediana	0.0308	0.0135	0.1719	0.0327	0.0625	0.0575	0.0467	0.0684
σ	0.0161	0.0083	0.3998	0.1404	0.0115	0.0138	0.0121	0.0107
Max	0.092	0.0245	1.3336	0.4451	0.0851	0.0774	0.0672	0.0864
Min	0.0079	0.0005	0.0001	0.0009	0.0391	0.0333	0.0199	0.0501
	UF5				UF6			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.068	0.0392	0.1004	0.3923	0.0895	0.0130	0.3107	0.6360
Mediana	0.066	0.0147	0.0496	0.2718	0.0834	0.0050	0.131	0.3914
σ	0.033	0.0503	0.1594	0.3140	0.0502	0.0205	0.4336	0.7071
Max	0.1385	0.1420	0.8418	1.0616	0.2171	0.0655	1.7741	1.9747
Min	0.0069	0.0000	0.0234	0.0743	0.0067	0.0000	0.1032	0.0262
	UF7				UF8			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.0093	0.0055	0.0932	0.0838	0.1444	0.1540	0.2404	0.3103
Mediana	0.0081	0.0030	0.0118	0.0969	0.1339	0.1532	0.3309	0.4760
σ	0.003	0.0061	0.1653	0.0396	0.0401	0.0371	0.1176	0.5593
Max	0.0171	0.0173	0.5472	0.1338	0.1968	0.2336	0.5387	1.3909
Min	0.0055	0.0003	0.0003	0.0231	0.0961	0.1004	0.0104	0.0208
	UF9				UF10			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.2149	0.2542	0.2955	0.0885	0.0832	0.0853	0.6685	0.1448
Mediana	0.2118	0.2327	0.2641	0.0439	0.0810	0.0593	0.6183	0.0527
σ	0.0650	0.0635	0.2285	0.1235	0.0551	0.0667	0.4177	0.2666
Max	0.3245	0.3564	1.0802	0.4114	0.2073	0.2065	1.7796	0.8640
Min	0.1249	0.1676	0.0834	0.0033	0.0004	0.0173	0.2639	0.0010

La Figura 6.8 y la Figura 6.9 proporcionan una manera cualitativa de comparar los enfoques discutidos. La primera compara visualmente el mejor conjunto de Pareto encontrado por cada algoritmo contra el Pareto-óptimo. Además, el conglomerado de soluciones se hace evidente en esta comparación. Ya que los resultados de un algoritmo estocástico no pueden ser juzgados solamente por la mejor solución que encontraron, la Figura 6.9 muestra las superficies de logro. Estos gráficos son útiles para visualizar el comportamiento de los algoritmos multiobjetivo a través de varias ejecuciones ya que muestran las mejores, medianas y peores superficies de logro. La convergencia mejorada obtenida por MOCCSA se origina en la combinación de los operadores simples de CSA y la longitud de vuelo caótica. Además, se observa una alta cobertura a través de los datos estadísticos de SP, MS e hipervolumen. Como se muestra en las figuras 6.8 y 6.9, la incorporación del caos a la longitud del vuelo mejora significativamente el rendimiento del algoritmo. Esto puede ser apreciado mediante la comparación de las gráficas presentes en las figuras 6.8 y 6.9 donde en el primer caso se observa como las mejores soluciones encontradas se aproximan al frente de Pareto óptimo, mientras que en el caso de la figura 6.9 se observan las superficies dominadas que cubren el espacio de soluciones. Estos resultados proporcionan evidencia de que MOCCSA se desempeña competitivamente cuando se aplica a un conjunto pruebas genéricas multiobjetivo.

Tabla 6.14. Resultados estadísticos de la MS para UF1 a UF10.

	UF1				UF2			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.8241	0.7320	0.7256	0.6766	0.9418	0.9165	0.8066	0.8235
Mediana	0.8439	0.7191	0.7551	0.6955	0.9627	0.9005	0.8106	0.8343
σ	0.0760	0.0597	0.1065	0.1055	0.0509	0.0435	0.0486	0.0320
Max	0.9299	0.8585	0.8167	0.8585	0.9935	0.9888	0.8799	0.8574
Min	0.7001	0.6246	0.4136	0.4586	0.8429	0.8504	0.6838	0.7445
	UF3				UF4			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.7217	0.6966	0.6210	0.6077	0.9843	0.9800	0.8634	0.8539
Mediana	0.7172	0.6979	0.6551	0.6087	0.9864	0.9830	0.8643	0.8554
σ	0.0267	0.0234	0.0755	0.0202	0.0108	0.0141	0.0129	0.0121
Max	0.7771	0.7314	0.7332	0.6382	0.9979	0.9983	0.8826	0.8756
Min	0.6820	0.6556	0.4919	0.5694	0.9608	0.9464	0.8358	0.8265
	UF5				UF6			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.7602	0.6233	0.6007	0.5162	0.7154	0.6845	0.6409	0.5392
Mediana	0.8168	0.6662	0.6623	0.5759	0.7231	0.7012	0.6647	0.5795
σ	0.1900	0.2813	0.2037	0.2105	0.1284	0.2106	0.0954	0.1697
Max	0.9367	0.9944	0.8175	0.7919	0.9338	0.8712	0.7825	0.7092
Min	0.1911	0.2194	0.2435	0.1536	0.5183	0.2900	0.4653	0.0000
	UF7				UF8			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.9782	0.7110	0.5670	0.6250	0.7828	0.7416	0.6658	0.6634
Mediana	0.9958	0.7242	0.5996	0.6339	0.8313	0.7739	0.6557	0.7083
σ	0.0213	0.0434	0.1579	0.0235	0.0480	0.0149	0.0116	0.0144
Max	0.9974	0.7708	0.8013	0.6526	0.9083	0.9305	0.7187	0.7811
Min	0.9442	0.6428	0.2653	0.5791	0.6645	0.8013	0.6330	0.6539
	UF9				UF10			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.6204	0.4500	0.3584	0.3948	0.7826	0.5688	0.4536	0.5000
Mediana	0.6333	0.4606	0.3788	0.4016	0.7966	0.5794	0.4797	0.5071
σ	0.0117	0.0246	0.0958	0.0135	0.0170	0.0347	0.1263	0.0188
Max	0.6365	0.4906	0.5076	0.4152	0.7979	0.6167	0.6410	0.5221
Min	0.6027	0.4057	0.1682	0.3677	0.7554	0.5142	0.2123	0.4633

Tabla 6.15. Resultados estadísticos del hipervolumen de UF1 a UF10.

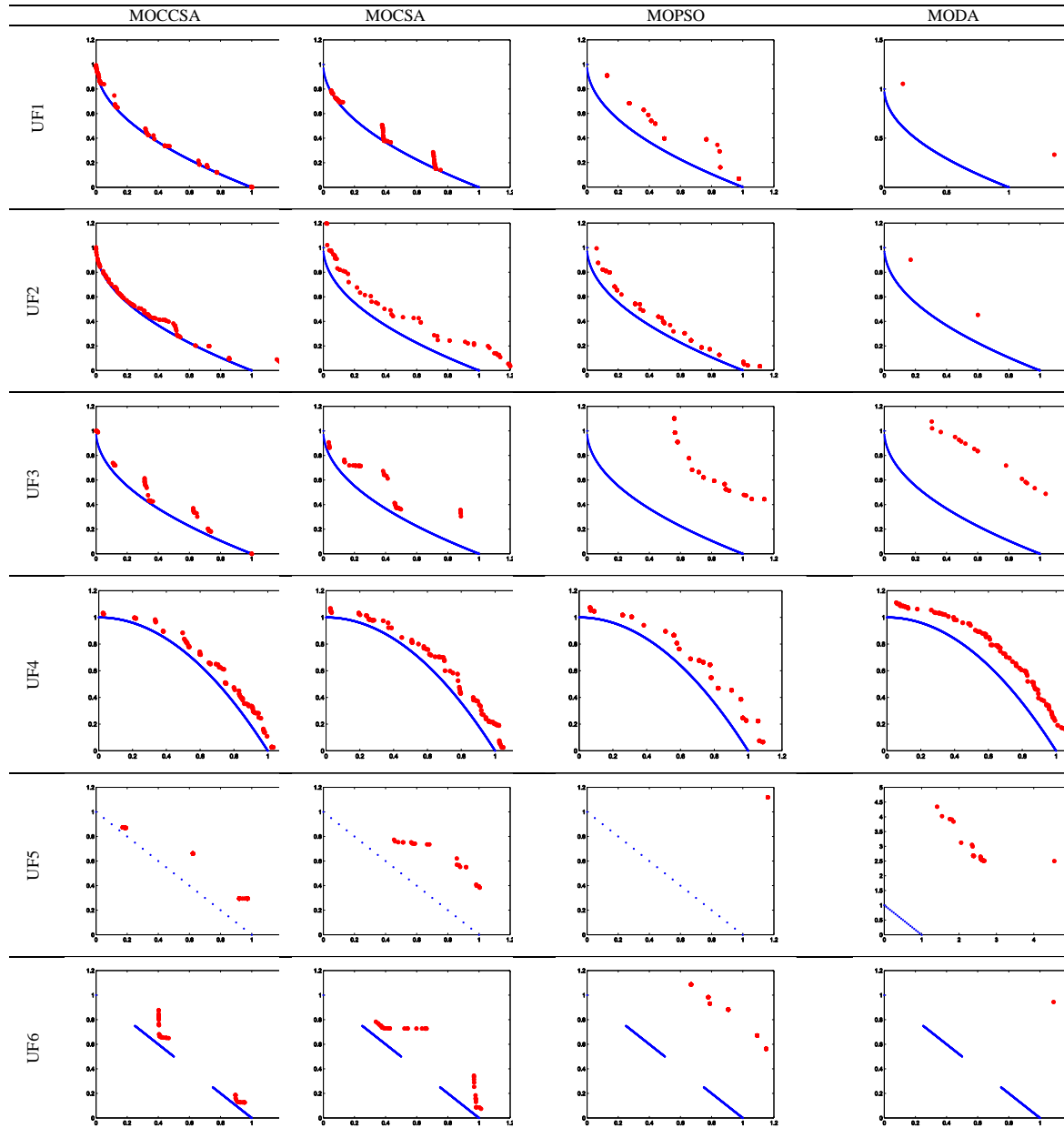
	UF1				UF2			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.9010	0.8910	0.8090	0.6705	0.9435	0.9270	0.8765	0.8465
Mediana	0.9100	0.8900	0.8150	0.6975	0.9450	0.9250	0.8725	0.8400
σ	0.0331	0.0312	0.0321	0.0748	0.0170	0.0254	0.0300	0.0315
Max	0.9450	0.9400	0.8600	0.7700	0.9650	0.9650	0.9250	0.9100
Min	0.8500	0.8300	0.7600	0.5250	0.9050	0.8900	0.8400	0.8100
	UF3				UF4			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.7620	0.7440	0.6490	0.6646	0.8155	0.8110	0.8555	0.8725
Mediana	0.7650	0.7525	0.6525	0.6061	0.8200	0.8125	0.8675	0.8750
σ	0.0392	0.0297	0.0523	0.0872	0.0255	0.0263	0.0356	0.0275
Max	0.8100	0.7650	0.7400	0.7834	0.8550	0.8400	0.8950	0.9200
Min	0.6800	0.6700	0.5650	0.5829	0.7700	0.7550	0.7900	0.8300
	UF5				UF6			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.7260	0.7035	0.1265	0.1971	0.7800	0.7660	0.2380	0.1235
Mediana	0.7250	0.7100	0.1275	0.1307	0.7950	0.7675	0.2175	0.1000
σ	0.0507	0.0777	0.0407	0.0684	0.0459	0.0646	0.0818	0.1095
Max	0.7950	0.7900	0.1950	0.1996	0.8600	0.8400	0.3800	0.3500
Min	0.5550	0.6150	0.0600	0.0697	0.7050	0.6400	0.1100	0.0050
	UF7				UF8			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.9180	0.7710	0.6960	0.6505	0.8340	0.8115	0.5940	0.6663
Mediana	0.9225	0.7750	0.6975	0.6650	0.8300	0.8150	0.5900	0.6495
σ	0.0315	0.0319	0.0344	0.0589	0.0380	0.0493	0.0694	0.0911
Max	0.9500	0.8250	0.7500	0.7350	0.9000	0.8900	0.6750	0.7600
Min	0.8400	0.7100	0.6200	0.5300	0.7800	0.7250	0.4750	0.4684
	UF9				UF10			
	MOCCSA	MOCSA	MOPSO	MODA	MOCCSA	MOCSA	MOPSO	MODA
Promedio	0.8140	0.8045	0.6385	0.6724	0.6100	0.5305	0.4581	0.4979
Mediana	0.8150	0.8150	0.6500	0.6782	0.5875	0.5200	0.4880	0.5740
σ	0.0425	0.0567	0.0385	0.1051	0.0314	0.0314	0.0450	0.0690
Max	0.9350	0.8550	0.6850	0.7545	0.7350	0.5950	0.5113	0.5598
Min	0.7300	0.7250	0.5850	0.5937	0.5150	0.4950	0.4069	0.4193

Dado que los métodos propuestos son muy similares, se necesita una prueba de significatividad para determinar si los resultados proporcionados por cada algoritmo no son lo suficientemente similares como para ser considerados como métodos diferentes. En este capítulo, la prueba de Wilcoxon se realiza con 35 muestras independientes que consideran un nivel de significancia del 5% sobre el valor de IGD correspondiente al frente de Pareto. La Tabla 6.16 presenta los valores p generados por la prueba de Wilcoxon para una comparación por pares del valor de IGD entre tres grupos formados por MOCCSA vs. MOCSA, MOCCSA vs. MOPSO, y MOCCSA vs. MODA.

Tabla 6.16. Resultados de la prueba estadística de Wilcoxon.

	MOCCSA vs MOCSA		MOCCSA vs MOPSO		MOCCSA vs MODA	
	Valor p	Hipótesis aceptada	Valor p	Hipótesis aceptada	Valor p	Hipótesis aceptada
UF1	2.9247E-11	1	2.9972E-11	1	2.6546E-11	1
UF2	2.9302E-11	1	3.0029E-11	1	1.2195E-11	1
UF3	2.9450E-11	1	3.0180E-11	1	6.2804E-12	1
UF4	2.9376E-11	1	3.0066E-11	1	2.5515E-11	1
UF5	2.9431E-11	1	3.0180E-11	1	2.7280E-11	1
UF6	2.1676E-01	0	3.0199E-11	1	2.3853E-11	1
UF7	2.9358E-11	1	3.0104E-11	1	2.2748E-11	1
UF8	6.9489E-01	0	2.9450E-11	1	9.4875E-12	1
UF9	2.0881E-05	1	2.9450E-11	1	1.9848E-11	1
UF10	5.9928E-01	0	2.9450E-11	1	2.1201E-11	1

Después de realizar la prueba de Wilcoxon, se observa en la Tabla 6.16 que MOCCSA es significativamente diferente de MOPSO y MODA. Esto es especialmente interesante, ya que los mecanismos multiobjetivo de MOPSO tales como el archivo, la rejilla adaptable y el mecanismo de selección de ruleta se utilizaron para generar MOCCSA. Además, el algoritmo MODA también utiliza esos mecanismos. Por otra parte, la comparación entre MOCSA y MOCCSA indica que por la sola inclusión de la serie caótica de Tent a un parámetro, la mayoría de las funciones de prueba reportan la presencia de una diferencia significativa que no puede ser atribuida a la casualidad, por lo que es posible afirmar que MOCSA y MOCCSA son algoritmos distintos estadísticamente.



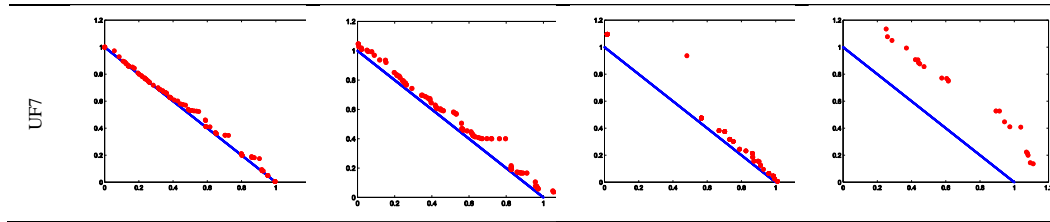
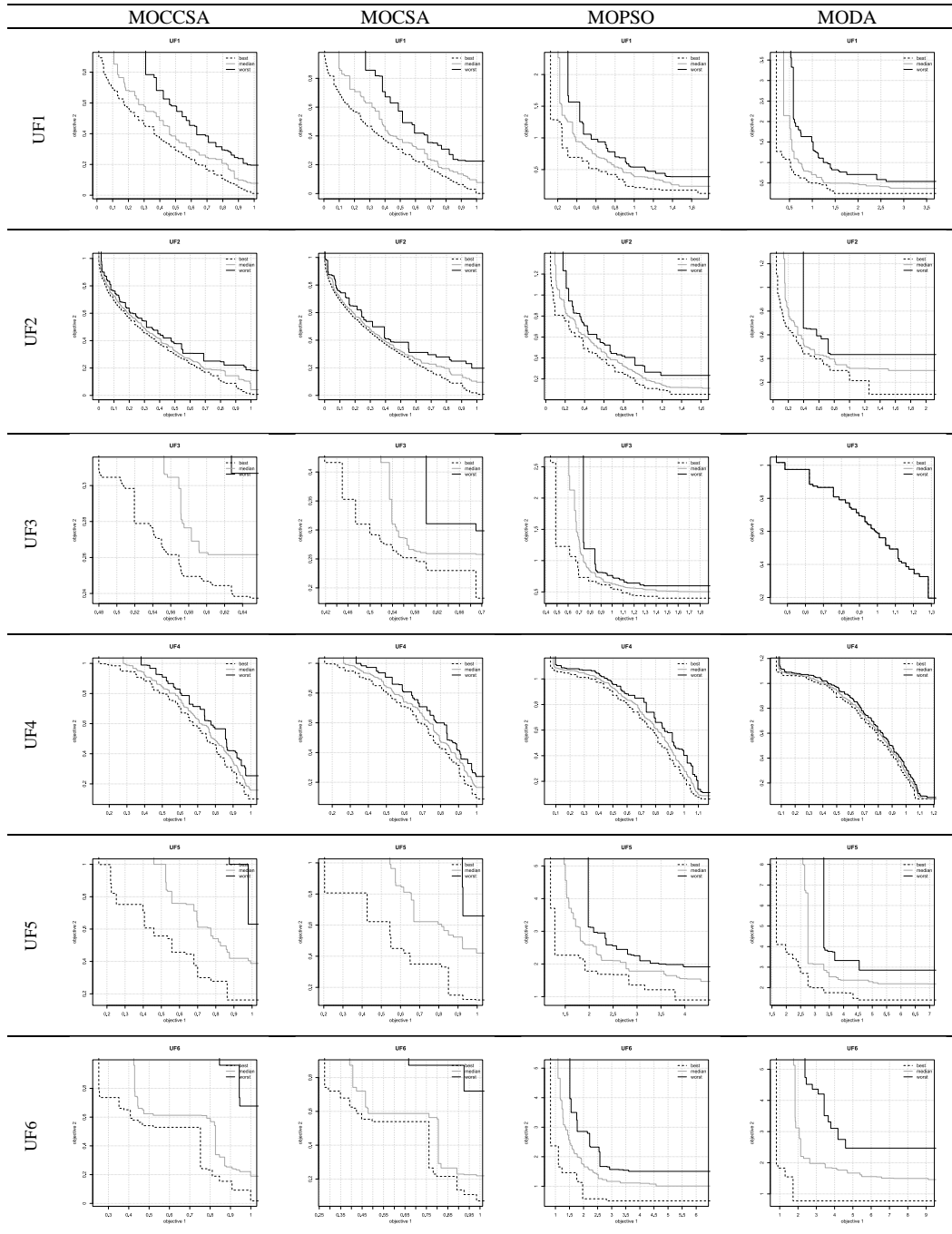


Figura 6.8. Comparación cualitativa, Pareto verdadero (azul) vs. mejor frente encontrado por cada algoritmo (rojo).



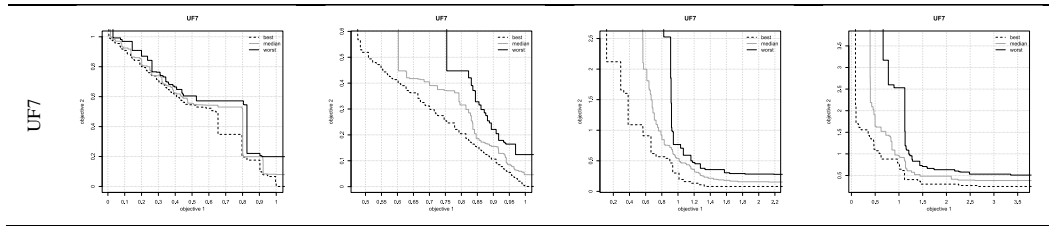


Figura 6.9. Superficies de logro de funciones de prueba bi-objetivo.

6.4 Resumen

En este capítulo se presentó un enfoque multiobjetivo para la umbralización de imágenes sin asistencia (UTH). De esta forma es posible resolver los dos problemas más importantes de la umbralización de imágenes como son la búsqueda de los valores de umbral y la elección de la cantidad óptima de umbrales. Esto se logra formulando el problema de UM como un problema de dos objetivos donde el primero se encarga de la calidad de la segmentación mientras que el segundo minimiza la cantidad de umbrales. Usando MOEAs es posible lograr un compromiso entre ambos objetivos que permita segmentar las imágenes adecuadamente. Además, en este capítulo se propuso un nuevo MOEA basado en el algoritmo CSA y series caóticas con resultados competitivos y prometedores.

7 Conclusiones y trabajo futuro

7.1 Conclusiones específicas

Esta memoria de tesis recoge la investigación desarrollada siguiendo cuatro líneas de investigación que contribuyen a la mejora de la segmentación de imágenes, donde la umbralización constituye un objetivo preferente. Dentro de estos planteamientos la optimización es una herramienta estudiada bajo distintos aspectos y perspectivas que mejora la segmentación y los procesos de umbralización. Las cuatro líneas mencionadas, junto con los capítulos donde se ha abordado la investigación, son: a) mejora de la calidad de la segmentación orientada a la umbralización utilizando entropía (capítulo tres); b) desempeño de algoritmos metaheurísticos en aplicaciones a imágenes de resonancia magnética y térmicas (capítulo cuatro); c) incorporación de información contextual como mejora de la umbralización (capítulo cinco) y d) optimización multiobjetivo para determinar el número de umbrales de forma automática (capítulo seis). En esta subsección se presentan las conclusiones específicas de cada una de esas líneas para después presentar las conclusiones generales.

7.1.1 Umbralización con entropía cruzada mínima.

En el capítulo tres se presenta un nuevo algoritmo para la segmentación multinivel basado en el algoritmo similar al electromagnetismo (EMO) para reducir las iteraciones y las evaluaciones de funciones. El enfoque propuesto considera el proceso de segmentación como un problema de optimización, empleando EMO para encontrar los puntos de umbral óptimos que minimizan la entropía cruzada (ECM). ECM combinado con EMO se centra directamente en la búsqueda del mejor conjunto de valores umbral. Este método considera sólo el histograma y el número de umbrales como entrada. Por el contrario, enfoques similares como las técnicas de

agrupación necesitan explorar las condiciones de inicialización que conducen a esquemas más complejos.

El algoritmo basado en EMO puede reducir sustancialmente el número de evaluaciones de funciones preservando las buenas capacidades de búsqueda de un método evolutivo. La técnica presentada es capaz de encontrar los mejores valores incluso con imágenes con altas resoluciones espaciales y complejas en cuanto al contenido de las mismas. En el enfoque desarrollado, el algoritmo utiliza partículas para codificar un conjunto de puntos de umbral candidatos. El ECM es la función objetiva, donde se evalúa la calidad de todas las soluciones candidatas. Las partículas se desarrollan utilizando la fuerza, la carga, el movimiento y los operadores de búsqueda local de EMO. Una vez obtenidos los umbrales óptimos, éstos se utilizan para segmentar la imagen. Para evaluar la calidad de las imágenes segmentadas, se propone el uso del PSNR, σ , SSIM y FSIM. Tales métricas consideran las coincidencias entre la imagen original y la segmentada.

El estudio experimental compara el enfoque propuesto con otros cinco enfoques relacionados, la Evolución Diferencial (DE), el algoritmo de Optimización del Enjambre de Partículas (PSO), la Búsqueda de Armonía (HS), el Algoritmo de la Araña Social (SSA), y la ampliamente utilizada Colonia Artificial de Abejas (ABC). La eficiencia de los algoritmos se evalúa en términos de PSNR, σ , SSIM, FSIM y valores de *fitness*. La alta tasa de convergencia de EMO es evidente en las comparaciones reportadas. Asimismo, EMO supera, en la mayoría de los experimentos, al resto de métodos con los que se compara proporcionando altas puntuaciones en las métricas evaluadas. Es posible establecer que el enfoque basado en EMO y ECM alcanza los valores mínimos de entropía cruzada en menos iteraciones y obtiene soluciones más precisas que DE, PSO, HS, SSA y ABC. Aunque los resultados ofrecen evidencia para demostrar que el método EMO puede dar buenos resultados en imágenes complejas, desde el punto de vista de su contenido, el objetivo de esta investigación no es diseñar un algoritmo de umbralización multinivel que pueda superar a todos los métodos disponibles actualmente, sino mostrar que los sistemas de electromagnetismo pueden ser considerados como una alternativa atractiva para este propósito.

Con el espíritu de contribuir a trabajos futuros, el desarrollo de técnicas de umbralización podría tomar tres caminos igualmente importantes. La primera es la tendencia actual, en la que los enfoques de umbralización se benefician de nuevas estrategias de optimización, especialmente si tales métodos mejoran el estado general de la técnica. Se puede tomar un segundo camino si el problema de los umbrales se resuelve con una metodología multiobjetivo. La incorporación de optimizadores multiobjetivo puede conducir a mejoras simultáneas en los criterios de umbralización. Este tema debería abordarse para mejorar aún más

la calidad de las imágenes segmentadas. En tercer lugar, la eficacia de los umbrales puede aplicarse directamente a áreas críticas de procesamiento de imágenes, como el procesamiento de imágenes médicas o satelitales. Cualquiera de los caminos mencionados contribuye significativamente a la comunidad de procesamiento de imágenes con nuevas metodologías.

7.1.2 Umbralización de imágenes especiales.

En la primera parte del capítulo cuatro se presenta una nueva metodología llamada ECM-CSA para la umbralización de imágenes cerebrales por Resonancia Magnética (RM). ECM-CSA se basa en un algoritmo evolutivo puesto a disposición de la comunidad científica. Para ello, se selecciona el problema de umbralización multinivel debido a su naturaleza de alta modalidad y su complejidad. La metodología propuesta considera el proceso de umbralización como un problema de optimización, donde CSA busca los puntos de umbral óptimos considerando la entropía cruzada como función objetivo. El CSA utiliza partículas para codificar un conjunto de puntos de umbral candidatos. La entropía cruzada se utiliza como función objetivo, evaluando la calidad de los puntos de umbral seleccionados. Siguiendo los valores de la función objetivo, los operadores de CSA guían el proceso de evolución a la vez que mejoran la segmentación de las imágenes cerebrales de RM. Para evaluar la calidad de las imágenes segmentadas se utilizan σ , PSNR, SSIM y FSIM. Estas métricas consideran la similitud entre las imágenes originales y las segmentadas.

La metodología propuesta fue validada usando dos conjuntos de imágenes de referencia; el primer conjunto está compuesto de imágenes generales comúnmente usadas en la literatura de procesamiento de imágenes, mientras que el segundo está formado de imágenes extraídas de la base de datos Brainweb. El primero tiene por objeto comparar el rendimiento de la CSA con alternativas bien conocidas. Para este fin, se implementan otros dos algoritmos evolutivos, la Evolución Diferencial (DE) y la Búsqueda de Armonías (HS). La comparación de los algoritmos se evalúa en relación con PSNR, σ , SSIM, FISM y valores de *fitness*. Tales comparaciones evidencian la convergencia, exactitud y robustez de la CSA, en contraste con las de la DE y HS. La prueba de Wilcoxon se utiliza para determinar que los resultados de la CSA son significativamente diferentes de los de DE y HS y no ocurrieron por casualidad. La segunda comparación tiene por objeto evaluar el rendimiento del ECM-CSA propuesto cuando se aplica específicamente a la segmentación de las imágenes cerebrales de RM. En este caso, los resultados cuantitativos indican que ECM-CSA genera imágenes segmentadas de RM de alta calidad con respecto a PSNR, SSIM y FSIM. Además, el análisis cualitativo de los resultados después de segmentar las imágenes cerebrales en los diferentes cortes muestra regiones bien delimitadas que son más fáciles de distinguir en comparación con otras técnicas.

A pesar de que los resultados ofrecen evidencia de que el método ECM-CSA propuesto funciona bien en las imágenes de RM, específicamente en la RM cerebral, este trabajo no está dirigido a diseñar un método de umbral multinivel capaz de superar todos los métodos actualmente disponibles, sino a evaluar el desempeño del método propuesto para procesar imágenes de RM en aplicaciones reales y no sólo en imágenes de prueba.

La segunda parte del capítulo cuatro analiza el problema de umbralización en las imágenes FLIR. Para ello, se implementan dos técnicas de umbralización clásicas basadas en el método de Otsu (1979) y la entropía Kapur et al. (1985). Ambas metodologías se utilizan como funciones objetivo en determinados EAs. ABC, DE, HS y FPA se utilizan para maximizar cada una de las funciones objetivo para dividir la imagen en varias clases. La comparación del rendimiento incluye dos enfoques: el primero compara los valores de aptitud para determinar qué algoritmo es el más adecuado para optimizar la función objetivo analizada comparando la aptitud media y la estabilidad media, mientras que el segundo enfoque se centra en la calidad de la imagen segmentada.

Considerando sólo los valores de *fitness* obtenidos para el método Otsu, FPA resulta ser la opción más adecuada de los algoritmos considerados en esta evaluación. Para el método de Kapur, ABC supera a los otros algoritmos. Sin embargo, los resultados indican que un buen valor de la función de *fitness* no se refleja necesariamente en una alta calidad de la imagen segmentada según las métricas PSNR FSIM y SSIM. Esta diferencia entre ambos resultados podría ser consecuencia de la naturaleza ruidosa de las imágenes FLIR y de la transición suave entre los valores de intensidad de las clases. Estas condiciones hacen que el procesamiento de imágenes FLIR sea una tarea difícil. De acuerdo con las métricas de calidad de las imágenes segmentadas, HS mostró un mejor desempeño en la mayoría de los experimentos tanto para Otsu como para Kapur. El algoritmo HS proporcionó excelentes resultados en cuanto a calidad y requirió menos tiempo computacional que ABC y FPA para realizar la optimización, convirtiéndolo en una alternativa atractiva para determinación de umbrales en imágenes FLIR. A pesar de que los resultados no mostraron un ganador absoluto para todas las pruebas e imágenes, los resultados obtenidos proporcionan información útil sobre el comportamiento de los EAs analizados aplicando los criterios de Otsu y Kapur para el problema de UM con las características desafiantes de las imágenes FLIR.

7.1.3 Umbralización basado en contexto.

En el capítulo cinco se presenta la implementación de dos algoritmos de optimización para el problema de la umbralización de imágenes multinivel. Estos dos enfoques han sido recientemente propuestos en la literatura relacionada; uno de ellos es el ALO que se inspira en el comportamiento de caza de las hormigas león, mientras que el SCA utiliza las funciones seno

y coseno para modificar las posiciones de las soluciones candidatas. En este trabajo, ALO y SCA se prueban sobre dos funciones objetivos (Otsu y Kapur) para encontrar los mejores umbrales para la segmentación de imágenes. A diferencia de otros enfoques similares, aquí también se considera la información espacial contextual de la imagen para seleccionar los umbrales. Este método utiliza lo que se conoce como curva de energía de la imagen, y tiene propiedades similares al histograma, pero también incluye información sobre la vecindad de cada píxel, que constituye el aporte clave fundamental de esta propuesta frente a otras basadas exclusivamente en el histograma, que sólo aporta información de los niveles de intensidad.

Los enfoques de umbralización propuestos basados en ALO y SCA han sido probados sobre un conjunto de imágenes de referencia. Además, considerando que ALO y SCA son enfoques relativamente novedosos, se han comparado con GA, PSO, CSA, RRA y ACOR. De los resultados experimentales, se puede concluir, con carácter general, que ALO supera en eficiencia al resto de los algoritmos con los que se compara. El proceso de segmentación en este trabajo es un problema multidimensional ya que se utilizan imágenes con representación 8 bits en los niveles de intensidad; el número de umbrales se establece en 2, 4, 8, 16 y 32. Basado en los resultados obtenidos, el trabajo futuro podría incluir la implementación de ALO con curva de energía para imágenes médicas, incluyendo RM, para el estudio de células sanguíneas por ejemplo o bien para imágenes térmicas con diferentes contenidos.

7.1.4 Segmentación de imágenes con un enfoque multiobjetivo

En el capítulo seis se presenta una nueva metodología llamada *Unassisted Multi-level Thresholding* (UTH). UTH trata con los dos problemas principales de la umbralización de imágenes; la determinación del número de umbrales, y la colocación óptima de los umbrales sobre el histograma de la imagen. La metodología propuesta considera el proceso de umbralización como un problema de optimización con dos objetivos: el primero es determinar el número mínimo óptimo de umbrales y el segundo es la selección óptima de los valores umbral que mejoran la calidad de la imagen segmentada. Estos dos objetivos son conflictivos. Dado que el número de umbrales puede tener un gran impacto en la calidad del resultado, un gran número de umbrales beneficia el resultado del proceso de segmentación, mientras que un pequeño número de umbrales genera resultados de baja calidad. Así pues, debería establecerse un equilibrio entre el número de umbrales y la calidad del resultado.

Las técnicas de optimización multiobjetivo abordan intrínsecamente los dos objetivos en conflicto, proporcionando mecanismos para identificar las mejores soluciones de compromiso entre dichos objetivos. UTH se aprovecha de tales operadores para obtener el umbral de imagen mientras que simultáneamente determina el número y la colocación de los umbrales sin ningún conocimiento previo o la intervención de un supervisor. En UTH, la

primera función objetivo está dedicada a maximizar la calidad de la imagen segmentada utilizando un criterio no paramétrico como es la entropía cruzada mínima, la ECS. La segunda función objetivo es el número de umbrales normalizado. Siguiendo los valores de cada función objetivo, los operadores de la MOEA seleccionada guían el proceso de evolución mejorando la calidad de la imagen segmentada.

El esquema propuesto se valida utilizando un conjunto de imágenes de referencia generadas a partir de imágenes comúnmente utilizadas en la literatura de procesamiento de imágenes. Dado que la UTH se propone como una metodología genérica más que como un algoritmo específico, su funcionamiento se evalúa en tres MOEA, a saber, NSGA-III, PESA-2 y MOPSO. El rendimiento de UTH se evalúa mediante la convergencia y un análisis de calidad. El análisis de convergencia se dedica a comparar la capacidad de cada implementación de UTH para encontrar un conjunto de soluciones competitivas. Esto se realiza analizando los valores de función objetivo, el hipervolumen y las superficies de logro. El análisis de calidad se realiza en relación con índices tales como PSNR, SSIM y FSIM para cada imagen segmentada. Estas comparaciones evidencian la convergencia y exactitud de la UTH propuesta, siendo la implementación de NSGA-III la más competitiva. A pesar de que los resultados ofrecen evidencia de que el método UTH-NSGA-III propuesto se desempeña bien en las imágenes de referencia, este trabajo no está dirigido a diseñar un método de umbral multinivel capaz de superar todos los métodos actualmente disponibles, sino a contribuir con la propuesta de un esquema general de segmentación no asistida para un umbral verdaderamente automático.

De forma paralela, el hecho de abordar la problemática de UM mediante MOEAs llevó a la propuesta de un nuevo MOEA. Así, en la segunda sección del capítulo seis, el recientemente propuesto algoritmo de búsqueda de cuervo se amplía para realizar una optimización multiobjetivo. Los operadores de CSA se mejoran tanto para las etapas de exploración como de explotación. La exploración se mejora incorporando una longitud de vuelo caótica. Dado que la CSA se propuso originalmente como una técnica de optimización de un único objetivo, es necesario añadir nuevos mecanismos a la CSA para considerar múltiples objetivos. La incorporación de un archivo para almacenar soluciones no dominadas unidas a una rejilla adaptativa destinada a mantener diversas soluciones llevó a la propuesta de MOCSA y MOCCSA.

Las metodologías propuestas se evaluaron utilizando un punto de referencia estándar consistente en 10 funciones desafiantes propuestas para el CEC (Q. Zhang et al., 2009). El desempeño de MOCCSA se contrasta con tres enfoques: MOCSA, su versión libre caótica; MOPSO, un algoritmo bien conocido; y MODA, un algoritmo de última generación. La competencia de los algoritmos se evalúa con respecto a IGD, SP, MS e hipervolumen. Los

resultados cualitativos y cuantitativos evidencian la convergencia y cobertura superior de MOCCSA. Debido a la similitud de MOCCSA y MOCSA se realizó una prueba de significancia para determinar si los resultados proporcionados por cada método son lo suficientemente diferentes como para ser considerados como dos algoritmos diferentes y no ocurrieron por casualidad. La prueba de Wilcoxon se aplicó a una comparación por pares de la métrica IGD reportada por cada algoritmo, y concluyó que la mayoría de los resultados son significativamente diferentes y no casuales, lo que indica que la incorporación de una longitud de vuelo caótica mejora el rendimiento del algoritmo.

A pesar de que los resultados ofrecen evidencia de que el método MOCCSA se desempeña bien en el punto de referencia seleccionado, este trabajo no pretende diseñar un método multicriterio capaz de superar todas las técnicas actualmente disponibles, sino analizar el efecto de añadir una perturbación caótica a un simple algoritmo multiobjetivo. Sin embargo, MOCCSA demuestra ser una metodología interesante que puede ser aplicada a todo tipo de campos de estudio, especialmente en problemas con frentes de Pareto difíciles.

7.2 Conclusiones generales

El desarrollo de la investigación se describe y concreta a lo largo de los capítulos donde se analizaron distintas problemáticas relacionadas con la umbralización de imágenes de cara a la segmentación. Las problemáticas se seleccionaron debido a los problemas identificados en la literatura relacionada. De esta forma se ha hecho énfasis en los cuatro aspectos reflejados en las conclusiones específicas.

Cada problemática ha sido abordada proponiendo una serie de métodos destinados a mejorar el estado del arte en cuanto a ese tema se refiere. Cada uno de los métodos propuestos se ha evaluado de acuerdo con un contexto apropiado haciendo énfasis en términos como la calidad de la segmentación y la capacidad de convergencia del método. Adicionalmente, cada método se evaluó en términos de significancia para asegurar que las propuestas sean significativas. Los resultados de los experimentos junto con sus respectivas comparaciones arrojan evidencia de que sugiere que los métodos propuestos son mejores que sus contrapartes evaluadas.

Además, es necesario mencionar que como resultado de los resultados obtenidos en cada uno de los temas abordados ha sido posible publicar en revistas internacionales de alto impacto para así cumplir con los objetivos de esta investigación.

7.3 Trabajo futuro

En esta investigación ha sido posible alcanzar todos los objetivos planteados al inicio de la investigación. Aun así, el estado del arte del procesamiento de imágenes cambia vertiginosamente en favor de nuevos enfoques, tal es el caso de los procesamientos basados en nuevos paradigmas para segmentación donde el aprendizaje profundo (*deep learning*), y dentro de él las redes neuronales convolucionales, constituye un aspecto digno de consideración (Dai, Li, He, & Sun, 2016; Davies, 2017). Bajo estas consideraciones los procesamientos asistidos por GPU aparecen como firmes candidatos. Es por ello que el campo de optimización debería adoptar implementaciones que permitan paralelizar sus operaciones para mantenerse competitivas ante las tecnologías emergentes.

Otra línea de investigación de vital importancia es la segmentación de imágenes médicas donde existen números métodos de adquisición de imágenes que suponen características y problemáticas propias de cada problema. Además, la optimización multiobjetivo supone un campo de investigación muy importante por sí solo. El hecho de replantear problemas clásicos mediante enfoques multiobjetivo puede aportar nuevas técnicas e ideas para resolver problemáticas que originalmente se abordan de forma individual.

Bibliografía

- Addis, B., Locatelli, M., & Schoen, F. (2005). Local optima smoothing for global optimization. *Optimization Methods and Software*, 20(4–5), 417–437.
<https://doi.org/10.1080/10556780500140029>
- Agrawal, S., Panda, R., Bhuyan, S., & Panigrahi, B. K. (2013). Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. *Swarm and Evolutionary Computation*, 11, 16–30. <https://doi.org/10.1016/j.swevo.2013.02.001>
- Akay, B., & Bahriye. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 13(6), 3066–3091.
<https://doi.org/10.1016/j.asoc.2012.03.072>
- Al-Kassir, A. R., Fernandez, J., Tinaut, F. V., & Castro, F. (2005). Thermographic study of energetic installations. *Applied Thermal Engineering*, 25(2–3), 183–190.
<https://doi.org/10.1016/j.applthermaleng.2004.06.013>
- Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers and Structures*, 169, 1–12.
<https://doi.org/10.1016/j.compstruc.2016.03.001>
- Aziz, M. A. El, Ewees, A. A., & Hassanien, A. E. (2017). Whale Optimization Algorithm and Moth-Flame Optimization for multilevel thresholding image segmentation. *Expert Systems with Applications*, 83, 242–256. Retrieved from
<http://www.sciencedirect.com/science/article/pii/S0957417417302671>
- Birbil, Ş. İ., & Fang, S.-C. (2003). An Electromagnetism-like Mechanism for Global Optimization. *Journal of Global Optimization*, 25(3), 263–282.
<https://doi.org/10.1023/A:1022452626305>

- Birbil, Ş. I., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 25(1), 263–282.
<https://doi.org/10.1023/A:1022452626305>
- Birbil, Ş. I., Fang, S. C., & Sheu, R. L. (2004). On the Convergence of a Population-Based Global Optimization Algorithm. *Journal of Global Optimization*, 30(2–3), 301–318.
- Black, H. (2013). Social Skills to Crow About. *Scientific American Mind*, 24(4), 12–12.
<https://doi.org/10.1038/scientificamericanmind0913-12>
- Blum, C., & Merkle, D. (2008). *Swarm intelligence : introduction and applications*. Springer.
- Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237, 82–117. <https://doi.org/10.1016/J.INS.2013.02.041>
- Brainweb. (2018). Retrieved September 17, 2018, from
<http://brainweb.bic.mni.mcgill.ca/brainweb>
- Caponetto, R., Fortuna, L., Fazzino, S., & Xibilia, M. G. (2003). Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(3), 289–304. <https://doi.org/10.1109/TEVC.2003.810069>
- Coello, C. A. C., Pulido, G. T. G. T., Lechuga, M. S. M. S., Coello Coello, C. A., Pulido, G. T. G. T., & Lechuga, M. S. M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 256–279.
<https://doi.org/10.1109/TEVC.2004.826067>
- Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11–12), 1245–1287.
[https://doi.org/10.1016/S0045-7825\(01\)00323-1](https://doi.org/10.1016/S0045-7825(01)00323-1)
- Corne, D., Jerram, N., Knowles, J. D., Oates, M., & Martin, J. (2001). PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, 283–290. <https://doi.org/citeulike-article-id:8133801>
- Corne, D. W., Knowles, J. D., & Oates, M. J. (2000). The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization (pp. 839–848). Springer, Berlin, Heidelberg.
https://doi.org/10.1007/3-540-45356-3_82
- Črepinšek, M., Liu, S.-H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms. *ACM Computing Surveys*, 45(3), 1–33.

<https://doi.org/10.1145/2480741.2480752>

- Cuevas, E., Díaz Cortés, M. A., & Oliva Navarro, D. A. (2016). A Swarm Global Optimization Algorithm Inspired in the Behavior of the Social-Spider (pp. 9–33). Springer, Cham.
https://doi.org/10.1007/978-3-319-28503-0_2
- Cuevas, E., González, M., Zaldivar, D., Pérez-Cisneros, M., & García, G. (2012). An Algorithm for Global Optimization Inspired by Collective Animal Behavior. *Discrete Dynamics in Nature and Society*, 2012, 1–24. <https://doi.org/10.1155/2012/638275>
- Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks, 379–387. Retrieved from <http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks>
- Davies, E. R. (2017). *Computer Vision, Fifth Edition: Principles, Algorithms, Applications, Learning*.
- De Castro, L. N., & Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3), 239–251.
<https://doi.org/10.1109/TEVC.2002.1011539>
- De Jong, K. (1988). Learning with genetic algorithms: An overview. *Machine Learning*, 3, 121–138. <https://doi.org/10.1007/BF00113894>
- Deb, K. (1999). Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3), 205–230.
<https://doi.org/10.1162/evco.1999.7.3.205>
- Deb, K., & Jain, H. (2013). An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *Ieeexplore.Ieee.Org*, 18(c), 1–1.
<https://doi.org/10.1109/TEVC.2013.2281534>
- Deb, K., Member, A., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm :, 6(2), 182–197.
- Deb, K., Pratab, S., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NGS-II. *IEEE Transactions on Evolutionary Computing*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Deb, K., & Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06* (p. 635). New York, New York, USA: ACM

Press. <https://doi.org/10.1145/1143997.1144112>

- Dehshibi, M. M., Sourizaei, M., Fazlali, M., Talaee, O., Samadyar, H., & Shanbehzadeh, J. (2017). A hybrid bio-inspired learning algorithm for image segmentation using multilevel thresholding. *Multimedia Tools and Applications*, 76(14). <https://doi.org/10.1007/s11042-016-3891-3>
- Djerou, L., Khelil, N., Dehimi, N. H., & Batouche, M. (2012). Automatic Multi-Level Thresholding Segmentation Based on Multi-Objective Optimization. *International Journal of Computer Science & Mathematics*, 13(6), 24–31.
- Dorigo, M., Maniezzo, V., & Coloni, a. (1996). The ant systems: optimization by a colony of cooperative agents. *IEEE Transactions on Man, Machine and Cybernetics-Part B*, 26(1).
- Erik, M., Pedersen, H., & Pedersen, M. E. H. (2010). Good parameters for particle swarm optimization. *Technical Report HL1001, Hvass Laboratories, HL1001*, 1–12. Retrieved from <http://www.hvass-labs.org/people/magnus/publications/pedersen10good-pso.pdf>
- Fausto, F., Cuevas, E., Valdivia, A., & González, A. (2017). A global optimization algorithm inspired in the behavior of selfish herds. *Biosystems*, 160, 39–55. <https://doi.org/10.1016/J.BIOSYSTEMS.2017.07.010>
- Fonseca, C. M., & Fleming, P. J. On the performance assessment and comparison of stochastic multiobjective optimizers, 1141 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) § (1996). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-61723-X_1022
- Gade, R., & Moeslund, T. B. (2014). Thermal cameras and applications: A survey. *Machine Vision and Applications*, 25(1), 245–262. <https://doi.org/10.1007/s00138-013-0570-5>
- García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC 2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15, 617–644. <https://doi.org/10.1007/s10732-008-9080-4>
- Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION* 76:2, 60–68.
- Geem, Z. W., Kim, J. H., Loganathan, G. V. V, Geem, Z. W., Kim, J. H., Loganathan, G. V. V, ... Loganathan, G. V. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, 76(2), 60–68. <https://doi.org/10.1177/003754970107600201>
- Ghosh, S., Bruzzone, L., Patra, S., Bovolo, F., & Ghosh, A. (2007). A Context-Sensitive

- Technique for Unsupervised Change Detection Based on Hopfield-Type Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3), 778–789.
<https://doi.org/10.1109/TGRS.2006.888861>
- Glover, F. (1989). Tabu Search—Part I. *ORSA Journal on Computing*, 1(3), 190–206.
<https://doi.org/10.1287/ijoc.1.3.190>
- Goldberg, D. E., & Holland, J. H. (1988). Genetic Algorithms and Machine Learning. *Machine Learning*, 3(2), 95–99. <https://doi.org/10.1023/A:1022602019183>
- Goldfeld, S. M., Quandt, R. E., & Trotter, H. F. (1966). Maximization by Quadratic Hill-Climbing. *Econometrica*, 34(3), 541. <https://doi.org/10.2307/1909768>
- Gonzalez, R. C., & Woods, R. E. (1992). *Digital Image Processing*. New Jersey: Pearson, Prentice-Hall.
- Hammouche, K., Diaf, M., & Siarry, P. (2008). A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Computer Vision and Image Understanding*, 109(2), 163–175. <https://doi.org/10.1016/j.cviu.2007.09.001>
- Hammouche, K., Diaf, M., & Siarry, P. (2010). A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. *Engineering Applications of Artificial Intelligence*, 23(5), 676–688. <https://doi.org/10.1016/j.engappai.2009.09.011>
- Hinojosa, S., Avalos, O., Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., & Gálvez, J. (2018). Unassisted thresholding based on multi-objective evolutionary algorithms. *Knowledge-Based Systems*, 159, 221–232. <https://doi.org/10.1016/j.knosys.2018.06.028>
- Hinojosa, S., Oliva, D., Cuevas, E., Pajares, G., Avalos, O., & Gálvez, J. (2018). Improving multi-criterion optimization with chaos: a novel Multi-Objective Chaotic Crow Search Algorithm. *Neural Computing and Applications*, 29(8), 319–335.
<https://doi.org/10.1007/s00521-017-3251-x>
- Hinojosa, S., Pajares, G., Cuevas, E., & Ortega-Sanchez, N. (2018). Thermal Image Segmentation Using Evolutionary Computation Techniques (pp. 63–88). Springer, Cham.
https://doi.org/10.1007/978-3-319-63754-9_4
- Horng, M.-H. (2010). Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. *Expert Systems with Applications*, 37(6), 4580–4592.
<https://doi.org/10.1016/j.eswa.2009.12.050>
- Horng, M.-H., & Liou, R.-J. (2011). Multilevel minimum cross entropy threshold selection based on the firefly algorithm. *Expert Systems with Applications*, 38(12), 14805–14811.

<https://doi.org/10.1016/j.eswa.2011.05.069>

- Hussein, W. A., Sahran, S., & Abdullah, S. N. H. S. (2016). A fast scheme for multilevel thresholding based on a modified bees algorithm. *Knowledge-Based Systems, 101*, 114–134. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0950705116300016#fig0002>
- Kamber, M., Shinghal, R., Collins, D. L., Francis, G. S., & Evans, A. C. (1995). Model-based 3-D segmentation of multiple sclerosis lesions in magnetic resonance brain images. *IEEE Transactions on Medical Imaging, 14*(3), 442–453. <https://doi.org/10.1109/42.414608>
- Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing, 29*(3), 273–285. [https://doi.org/10.1016/0734-189X\(85\)90125-2](https://doi.org/10.1016/0734-189X(85)90125-2)
- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Computer Engineering Department, Engineering Faculty, Erciyes University*.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- Kasprzak, E. M., & Lewis, K. E. (2000). An approach to facilitate decision tradeoffs in pareto solution sets. *Journal of Engineering Valuation and Cost Analysis, 3*(1), 173–187.
- Keck, J. D. and M. (2005). A two-stage approach to person detection in thermal imagery. *IEEE Workshop on Applications of Computer Vision*, (January).
- Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference On, 4*, 1942–1948 vol.4. <https://doi.org/10.1109/ICNN.1995.488968>
- Kittler, J., & Illingworth, J. (1986). Minimum error thresholding. *Pattern Recognition, 19*(1), 41–47. [https://doi.org/10.1016/0031-3203\(86\)90030-0](https://doi.org/10.1016/0031-3203(86)90030-0)
- Kullback, S. (1968). *Information Theory and Statistics*. New York: Dover.
- Kumar, S., Kumar, P., Sharma, T. K., & Pant, M. (2013). Bi-level thresholding using PSO, Artificial Bee Colony and MRLDE embedded with Otsu method. *Memetic Computing, 5*(4), 323–334. <https://doi.org/10.1007/s12293-013-0123-5>
- Leykin, A., & Hammoud, R. (2010). Pedestrian tracking by fusion of thermal-visible surveillance videos. *Machine Vision and Applications, 21*(4), 587–595. <https://doi.org/10.1007/s00138-008-0176-5>

- Li, C. H., & Lee, C. K. (1993). Minimum cross entropy thresholding. *Pattern Recognition*, 26(4), 617–625. [https://doi.org/10.1016/0031-3203\(93\)90115-D](https://doi.org/10.1016/0031-3203(93)90115-D)
- Li, L., Sun, L., Guo, J., Qi, J., Xu, B., & Li, S. (2017). Modified Discrete Grey Wolf Optimizer Algorithm for Multilevel Image Thresholding. *Computational Intelligence and Neuroscience*, 2017, 1–16. <https://doi.org/10.1155/2017/3295769>
- Liao, P. S., Chen, T. S., & Chung, P. C. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, 17(5), 713–727. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-0035440214&partnerID=tZOtx3y1>
- Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, 76(2), 60–68. <https://doi.org/10.1177/003754970107600201>
- López-Ibáñez, M., Paquete, L., & Stützle, T. (2010). Exploratory Analysis of Stochastic Local Search Algorithms in Biobjective Optimization. In *Experimental Methods for the Analysis of Optimization Algorithms* (pp. 209–222). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02538-9_9
- Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated Local Search. In *Handbook of Metaheuristics* (pp. 320–353). Boston: Kluwer Academic Publishers. https://doi.org/10.1007/0-306-48056-5_11
- Mantegna, R. (1994). Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Physical Review E*. Retrieved from <http://journals.aps.org/pre/abstract/10.1103/PhysRevE.49.4677>
- MathWorks. (2018). Matlab 8.3. Retrieved from <https://es.mathworks.com/>
- Merrikh-Bayat, F. (2015). The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. *Applied Soft Computing*, 33, 292–303. <https://doi.org/10.1016/J.ASOC.2015.04.048>
- Mirjalili, S. (2015). The Ant Lion Optimizer. *Advances in Engineering Software*, 83, 80–98. <https://doi.org/10.1016/J.ADVENGSOFT.2015.01.010>
- Mirjalili, S. (2016a). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4), 1053–1073. <https://doi.org/10.1007/s00521-015-1920-1>
- Mirjalili, S. (2016b). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120–133. <https://doi.org/10.1016/j.knosys.2015.12.022>
- Mirjalili, S. (2019). *Evolutionary Algorithms and Neural Networks* (Vol. 780). Studies in

- Computational Intelligence. <https://doi.org/10.1007/978-3-319-93025-1>
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163–191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- Mirjalili, S., Jangir, P., Mirjalili, S. Z., Saremi, S., & Trivedi, I. N. (2017). Optimization of problems with multiple objectives using the multi-verse optimization algorithm. *Knowledge-Based Systems*, 134, 50–71. <https://doi.org/10.1016/j.knosys.2017.07.018>
- Moeskops, P., Viergever, M. A., Mendrik, A. M., De Vries, L. S., Benders, M. J. N. L., & Išgum, I. (2016). Automatic Segmentation of MR Brain Images with a Convolutional Neural Network. *IEEE Transactions on Medical Imaging*, 35(5), 1252–1261. <https://doi.org/10.1109/TMI.2016.2548501>
- Nawandhar, A. A., Yamujala, L., & Kumar, N. (2015). Image segmentation using thresholding for cell nuclei detection of colon tissue. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1199–1203). IEEE. <https://doi.org/10.1109/ICACCI.2015.7275775>
- Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., & Perez-Cisneros, M. (2013). Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*, 2013. <https://doi.org/10.1155/2013/575414>
- Oliva, D., Hinojosa, S., Cuevas, E., Pajares, G., Avalos, O., & Gálvez, J. (2017). Cross entropy based thresholding for magnetic resonance brain images using Crow Search Algorithm. *Expert Systems with Applications*, 79, 164–180. <https://doi.org/10.1016/j.eswa.2017.02.042>
- Oliva, D., Hinojosa, S., & Demeshko, M. V. (2017). Engineering applications of metaheuristics: An introduction. *Journal of Physics: Conference Series*, 803(1), 012111. <https://doi.org/10.1088/1742-6596/803/1/012111>
- Oliva, D., Hinojosa, S., Elaziz, M. A., & Ortega-Sánchez, N. (2018). Context based image segmentation using antlion optimization and sine cosine algorithm. *Multimedia Tools and Applications*, 77(19), 25761–25797. <https://doi.org/10.1007/s11042-018-5815-x>
- Oliva, D., Hinojosa, S., Osuna-Enciso, V., Cuevas, E., Pérez-Cisneros, M., & Sanchez-Ante, G. (2017). Image segmentation by minimum cross entropy using evolutionary methods. *Soft Computing*, pp. 1–20. <https://doi.org/10.1007/s00500-017-2794-1>
- Omran, M. G. H., Salman, A., & Engelbrecht, A. P. (2006). Dynamic clustering using particle

- swarm optimization with application in image segmentation. *Pattern Analysis and Applications*, 8(4), 332–344. <https://doi.org/10.1007/s10044-005-0015-5>
- Ortiz, A., Gorriz, J. M., Ramírez, J., Salas-González, D., & Llamas-Elvira, J. M. (2013). Two fully-unsupervised methods for MR brain image segmentation using SOM-based strategies. *Applied Soft Computing Journal*, 13(5), 2668–2682. <https://doi.org/10.1016/j.asoc.2012.11.020>
- Osuna-Enciso, V., Cuevas, E., & Sossa, H. (2013). A comparison of nature inspired algorithms for multi-threshold image segmentation. *Expert Systems with Applications*, 40(4), 1213–1219. <https://doi.org/10.1016/j.eswa.2012.08.017>
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
- Pajares Martinsanz, G., & Cruz García, J. M. de la. (2007). *Visión por computador : imágenes digitales y aplicaciones*. Ra-Ma. Retrieved from <https://www.casadellibro.com/libro-vision-por-computador-imagenes-digitales-y-aplicaciones/9788478978311/1226163>
- Pal, N. (1996). On minimum cross-entropy thresholding. *Pattern Recognition*, 29(4), 575–580. [https://doi.org/10.1016/0031-3203\(95\)00111-5](https://doi.org/10.1016/0031-3203(95)00111-5)
- Pare, S., Bhandari, A. K., Kumar, A., & Singh, G. K. (2017). An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix. *Expert Systems with Applications*, 87, 335–362. <https://doi.org/10.1016/J.ESWA.2017.06.021>
- Pare, S., Kumar, A., Bajaj, V., & Singh, G. K. (2016). A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve. *Applied Soft Computing*, 47, 76–102. <https://doi.org/10.1016/J.ASOC.2016.05.040>
- Parsopoulos, K. E. (2002). Particle Swarm Optimization Method in Multiobjective Problems, 3, 603–607.
- Patra, S., Gautam, R., & Singla, A. (2014). A novel context sensitive multilevel thresholding for image segmentation. *Applied Soft Computing Journal*, 23, 122–127. <https://doi.org/10.1016/j.asoc.2014.06.016>
- Pitarma, R., Crisóstomo, J., & Jorge, L. (2016). Analysis of Materials Emissivity Based on Image Software (pp. 749–757). Springer, Cham. https://doi.org/10.1007/978-3-319-31232-3_70
- Portes de Albuquerque, M., Esquef, I. A., & Gesualdi Mello, A. R. (2004). Image thresholding

- using Tsallis entropy. *Pattern Recognition Letters*, 25(9), 1059–1065.
<https://doi.org/10.1016/j.patrec.2004.03.003>
- Ring, E. F. J., Ammer, K., & Ring, E. F. J. (2012). Infrared thermal imaging in medicine. *Physiological Measurement*, 33(3), R33. <https://doi.org/10.1088/0967-3334/33/3/R33>
- Rocha, A. M. a. C., & Fernandes, E. M. G. P. (2009). Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems. *International Journal of Computer Mathematics*, 86(November), 1932–1946.
<https://doi.org/10.1080/00207160902971533>
- Rocha, A. M. A. C., & Fernandes, E. M. G. P. (2009). Modified movement force vector in an electromagnetism-like mechanism for global optimization. *Optimization Methods and Software*, 24(2), 253–270.
- Rovcanin, M., De Poorter, E., Van Den Akker, D., Moerman, I., Demeester, P., & Blondia, C. (2014). Experimental validation of a reinforcement learning based approach for a service-wise optimisation of heterogeneous wireless sensor networks. *Wireless Networks*, 931–948. <https://doi.org/10.1007/s11276-014-0817-8>
- Sahoo, P. ., Soltani, S., & Wong, A. K. . (1988). A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41(2), 233–260. [https://doi.org/10.1016/0734-189X\(88\)90022-9](https://doi.org/10.1016/0734-189X(88)90022-9)
- Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1), 146. <https://doi.org/10.1117/1.1631315>
- Sarkar, S., Das, S., & Chaudhuri, S. S. (2016). Hyper-spectral image segmentation using Rényi entropy based multi-level thresholding aided with differential evolution. *Expert Systems with Applications*, 50, 120–129. <https://doi.org/10.1016/j.eswa.2015.11.016>
- Sathya, P. D., & Kayalvizhi, R. (2011). Optimal multilevel thresholding using bacterial foraging algorithm. *Expert Systems with Applications*, 38(12), 15549–15564.
<https://doi.org/10.1016/j.eswa.2011.06.004>
- Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13. Retrieved from <http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.1631315>
- Shilane, D., Martikainen, J., Dudoit, S., & Ovaska, S. J. (2008). A general framework for statistical performance comparison of evolutionary computation algorithms. *Information Sciences*, 178(14), 2870–2879. <https://doi.org/10.1016/j.ins.2008.03.007>

- Sierra, M. R., & Coello Coello, C. A. (2005). Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and E-Dominance. *Lecture Notes in Computer Science*, 3410, 505–519. https://doi.org/10.1007/978-3-540-31880-4_35
- Silva, L. F.; Saade, D. C. M.; Sequeiros, G. O.; Silva, A. C.; Paiva, A. C.; Bravo, R. S.; Conci, A. (2014). A New Database for Breast Research with Infrared Image. *Journal of Medical Imaging and Health Informatics*, 4, 92–100.
- Socha, K., & Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3), 1155–1173. <https://doi.org/10.1016/j.ejor.2006.06.046>
- Srinivas, N., & Deb, K. (1995). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3), 221–248. <https://doi.org/10.1162/evco.1994.2.3.221>
- Storn, R., & Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*.
- Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>
- Suman, B. (2005). Study of self-stopping PDMOSA and performance measure in multiobjective optimization. *Computers and Chemical Engineering*, 29(5), 1131–1147. <https://doi.org/10.1016/j.compchemeng.2004.12.002>
- Suresh, S., & Lal, S. (2017). Multilevel thresholding based on Chaotic Darwinian Particle Swarm Optimization for segmentation of satellite images. *Applied Soft Computing*, 55, 503–522. <https://doi.org/10.1016/j.asoc.2017.02.005>
- Suzuki, H., & Toriwaki, J. ichiro. (1991). Automatic segmentation of head MRI images by knowledge guided thresholding. In *Computerized Medical Imaging and Graphics* (Vol. 15, pp. 233–240). Pergamon. [https://doi.org/10.1016/0895-6111\(91\)90081-6](https://doi.org/10.1016/0895-6111(91)90081-6)
- Talbi, E.-G. (2002). A Taxonomy of Hybrid Metaheuristics. *Journal of Heuristics*, 8(5), 541–564. <https://doi.org/10.1023/A:1016540724870>
- Tang, K., Xiao, X., Wu, J., Yang, J., & Luo, L. (2017). An improved multilevel thresholding approach based modified bacterial foraging optimization. *Applied Intelligence*, 46(1), 214–226. <https://doi.org/10.1007/s10489-016-0832-9>
- Tang, K., Yuan, X., Sun, T., Yang, J., & Gao, S. (2011). An improved scheme for minimum

- cross entropy threshold selection based on genetic algorithm. *Knowledge-Based Systems*, 24(8), 1131–1138. <https://doi.org/10.1016/j.knosys.2011.02.013>
- USC-SIPI Image Database. (1981). USC-SIPI Image Database. Retrieved September 24, 2018, from <http://sipi.usc.edu/database/database.php>
- Vadivambal, R., & Jayas, D. S. (2011, February 24). Applications of Thermal Imaging in Agriculture and Food Industry-A Review. *Food and Bioprocess Technology*. Springer-Verlag. <https://doi.org/10.1007/s11947-010-0333-5>
- van Laarhoven, P. J. M., & Aarts, E. H. L. (1987). Simulated annealing. In *Simulated Annealing: Theory and Applications* (pp. 7–15). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-015-7744-1_2
- Vollmer, M., Möllmann, K.-P., & Wiley InterScience (Online service). (2010). *Infrared thermal imaging : fundamentals, research and applications*. Wiley-VCH.
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. <https://doi.org/10.1109/TIP.2003.819861>
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- Wu, P., Yang, W.-H., & Wei, N.-C. (2004). An Electromagnetism Algorithm of Neural Network Analysis—an Application To Textile Retail Operation. *Journal of the Chinese Institute of Industrial Engineers*, 21(1), 59–67. <https://doi.org/10.1080/10170660409509387>
- Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm (pp. 65–74). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12538-6_6
- Yang, X.-S., Karamanoglu, M., & He, X. (2014). Flower pollination algorithm: A novel approach for multiobjective optimization. *Engineering Optimization*, 46(9), 1222–1237. <https://doi.org/10.1080/0305215X.2013.832237>
- Yang, X. (2010). Engineering Optimization. In *Engineering Optimization* (pp. 15–28). NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470640425.ch2>
- Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2), 78. <https://doi.org/10.1504/IJBIC.2010.032124>

- Yang, X. S. (2012). Flower pollination algorithm for global optimization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7445 LNCS, 240–249. https://doi.org/10.1007/978-3-642-32894-7_27
- Yin, P.-Y. (2007). Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Applied Mathematics and Computation*, 184(2), 503–513. <https://doi.org/10.1016/j.amc.2006.06.057>
- Yu, J. J. Q., & Li, V. O. K. (2015). A social spider algorithm for global optimization. *Applied Soft Computing*, 30, 614–627. <https://doi.org/10.1016/j.asoc.2015.02.014>
- Zaitoun, N. M., & Aqel, M. J. (2015). Survey on Image Segmentation Techniques. *Procedia Computer Science*, 65, 797–806. <https://doi.org/10.1016/j.procs.2015.09.027>
- Zhang, D. (2011). FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(8), 2378–2386. <https://doi.org/10.1109/TIP.2011.2109730>
- Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition, 11(6), 712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., & Liu, W. (2009). Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. *Technical Report*, 1–30.
- Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8), 1335–1346. [https://doi.org/10.1016/0031-3203\(95\)00169-7](https://doi.org/10.1016/0031-3203(95)00169-7)
- Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2012). Decomposition-Based Multiobjective Evolutionary Algorithm With an Ensemble of Neighborhood Sizes. *IEEE Transactions on Evolutionary Computation*, 16(3), 442–446. <https://doi.org/10.1109/TEVC.2011.2166159>
- Zheng, Y., Zhao, F., Liu, H., & Wang, J. (2016). Adaptive Multi-level Thresholding Segmentation Based on Multi-objective Evolutionary Algorithm (pp. 606–615). Springer, Cham. https://doi.org/10.1007/978-3-319-41000-5_61
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32–49. <https://doi.org/10.1016/j.swevo.2011.03.001>
- Zitzler, E. (1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. *Ph.D. Thesis*, (30), 132. <https://doi.org/citeulike-article-id:4597043>

- Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *Eidgenössische Technische Hochschule Zürich (ETH), Institut Für Technische Informatik Und Kommunikationsnetze (TIK)*.
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271. <https://doi.org/10.1109/4235.797969>
- Zukal, M., Mekyska, J., Cika, P., & Smekal, Z. (2013). Interest points as a focus measure in multi-spectral imaging. *Radioengineering*, 22(1), 68–81. <https://doi.org/10.1109/TSP.2012.6256402>